


# Restricted Boltzmann machine: Recent advances and mean-field theory

Aurélien Decelle† and Cyril Furtlehner

**Citation:**Chin. Phys. B, 2021, 30 (**4**): 040202. DOI: 10.1088/1674-1056/abd160
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

**What follows is a list of articles you may be interested in**

---

## Dynamical analysis for hybrid virus infection system in switching environment

Dong-Xi Li(李东喜), Ni Zhang(张妮)
Chin. Phys. B, 2020, 29 (**9**): 090201. DOI: 10.1088/1674-1056/ab8c3f

## Stochastic responses of tumor—immune system with periodic treatment

Dong-Xi Li(李东喜), Ying Li(李颖)
Chin. Phys. B, 2017, 26 (**9**): 090203. DOI: 10.1088/1674-1056/26/9/090203

## Smoothing potential energy surface of proteins by hybrid coarse grained approach

Yukun Lu(卢禹锟), Xin Zhou(周昕), ZhongCan OuYang(欧阳钟灿)
Chin. Phys. B, 2017, 26 (**5**): 050202. DOI: 10.1088/1674-1056/26/5/050202

## Stochastic response of van der Pol oscillator with two kinds of fractional derivatives under Gaussian white noise excitation

Yong-Ge Yang(杨勇歌), Wei Xu(徐伟), Ya-Hui Sun(孙亚辉), Xu-Dong Gu(谷旭东)
Chin. Phys. B, 2016, 25 (**2**): 020201. DOI: 10.1088/1674-1056/25/2/020201

## Light trapping characteristics of glass substrate with hemisphere pit arrays in thin film Si solar cells

Chen Le, Wang Qing-Kang, Wangyang Pei-Hua, Huang Kun, Shen Xiang-Qian
Chin. Phys. B, 2015, 24 (**4**): 040202. DOI: 10.1088/1674-1056/24/4/040202

---

TOPICAL REVIEW — Machine learning in statistical physics

# Restricted Boltzmann machine: Recent advances and mean-field theory*

Aurélien Decelle[1,2,†] and Cyril Furtlehner[2]

[1] *Departamento de Física Téorica I, Universidad Complutense*, 28040 *Madrid, Spain*
[2] *TAU team INRIA Saclay & LISN Université Paris Saclay, Orsay* 91405, *France*

(Received 30 September 2020; revised manuscript received 13 November 2020; accepted manuscript online )

This review deals with restricted Boltzmann machine (RBM) under the light of statistical physics. The RBM is a classical family of machine learning (ML) models which played a central role in the development of deep learning. Viewing it as a spin glass model and exhibiting various links with other models of statistical physics, we gather recent results dealing with mean-field theory in this context. First the functioning of the RBM can be analyzed via the phase diagrams obtained for various statistical ensembles of RBM, leading in particular to identify a compositional phase where a small number of features or modes are combined to form complex patterns. Then we discuss recent works either able to devise mean-field based learning algorithms; either able to reproduce generic aspects of the learning process from some ensemble dynamics equations or/and from linear stability arguments.

**Keywords:** restricted Boltzmann machine (RBM), machine learning, statistical physics

**PACS:** 02.50.−r, 02.30.Z, 05.70.Fh **DOI:** 10.1088/1674-1056/abd160

## 1. Introduction

During the last decade, machine learning has experienced a rapid development, both in everyday life with the incredible success of image recognition used in various applications, and in research[1,2] where many different communities are now involved. This common effort involves fundamental aspects such as why it works or how to build new architectures and at the same time a search for new applications of machine learning to other fields, like for instance improving biomedical images segmentation[3] or detecting automatically phase transitions in physical systems.[4] Machine learning classical tasks are divided into at least two big categories: supervised and unsupervised learning (putting aside reinforcement learning and the more recently introduced approach of self-supervised learning). Supervised learning consists in learning a specific task — for instance recognizing an object on an image or a word in a speech — by giving the machine a set of samples together with the correct answer and correcting the prediction of the machine by minimizing a well-design and easy computable loss function. Unsupervised learning consists in learning a representation of the data given an explicit or implicit probability distribution, hence adjusting a likelihood function on the data. In this latter case, no label is assigned to the data and the result depends thus solely on the structure of the considered model and of the dataset.

In this review, we are interested in a particular model: the restricted Boltzmann machine (RBM). Originally called Harmonium[5] or product of experts,[6] RBMs were designed[7] to perform unsupervised tasks even though they can also be used to accomplish supervised learning in some sense. RBMs are part of what is called generative models which aim to learn a latent representation of the data in order to later be used to generate statistically similar new data — but different from those of the training set. There are Markov random fields (or Ising model for physicists), that were designed as a way to automatically interpret an image using a parallel architecture including a direct encoding of the probability of each "hypothesis" (latent description of a small portion of an image). Later on, RBMs started to take an important role in the machine learning (ML) community, when a simple learning algorithm introduced by Hinton *et al.*,[6] the contrastive divergence (CD), managed to learn a non-trivial dataset such as MNIST.[8] It was in the same period that RBMs became very popular in the ML community for its capability to pretrain deep neural networks (for instance deep auto-encoder), in a layer wise style. And, it was then showed that RBMs are universal approximator[9] of discrete distributions, that is, an arbitrary large RBM can approximate arbitrarily well any discrete distribution (which led to many rigorous results about the modelization mechanism of RBMs[10]). In addition, RBMs offer the possibility to be stacked to form a multi-layer generative model known as a deep Boltzmann machine (DBM).[11] In the more recent years, RBMs continued to attract scientific interest. Firstly because it can be used on continuous or discrete

*AD was supported by the Comunidad de Madrid and the Complutense University of Madrid (Spain) through the Atracción de Talento program (Ref. 2019-T1/TIC-13298).

†Corresponding author. E-mail: adecelle@ucm.es

© 2021 Chinese Physical Society and IOP Publishing Ltd 

variable very easily.[12–15] Secondly, because the possible interpretations of the hidden nodes can be very useful.[16,17] Interestingly, in some cases, more elaborate methods such as GAN[18] are not working better.[19] Finally it can be used for other tasks as well, such as classification or representation learning.[20] Besides all these positive aspects, the learning process itself of the RBM remains poorly understood. The reasons are twofold: firstly, the gradient can be computed only in an approximated way as we will see; secondly, simple changes may have terrible impact on the learning or, messed up completely with the other meta-parameters. For instance making a naive change of variable in the MNIST dataset[21,22] can affect importantly the training performance (In MNIST, it is usual to consider binary variable $\{0,1\}$ to describe the dataset. Taking instead $\{\pm 1\}$ naively will affect dramatically the learning of the RBM). In another example, varying the number of hidden nodes, while keeping the other mete-parameters fixed, will affect not only the representational power of the RBM but also the learning dynamics itself.

The statistical physics community, on its side, has a long tradition of studying inference and learning process with its own tools. Using idealized inference problems, it has managed in the past to shed light on the learning process of many ML models. For instance, in the Hopfield model,[23–26] a retrieval phase was characterized where the maximum number of patterns that can be retrieved can be expressed as a function of the temperature. Another example is the computation of the storage capacity of the Perceptron[27] on synthetic datasets.[28,29] In these approaches, the formalism of statistical physics explains the macroscopic behavior of the model in term of its position on a phase diagram in the large size limit.

From a purely technical point of view, the RBM can be seen for a physicist as a disordered Ising model on a bipartite graph. Yet, the difference with respect to the usual models that are studied in statistical physics is that the phase diagram of a trained RBM involves a highly non-trivial coupling matrix where the components are correlated as a result of the learning process. These dependencies make it non-trivial to adapt classical tools from statistical mechanics, such as the replica theory.[30] We will illustrate in this article how methods from statistical physics still have helped to characterize both the equilibrium phase of an idealized RBM where the coupling matrix has a structured spectrum, and how the learning dynamics can be analyzed in some specific regimes, both results being obtained with traditional mean-field approaches.

The paper is organized as follows. We will first give the definition of the RBM and review the typical learning algorithm used to train the model in Section 2, Then, in Section 3, we will review different types of RBMs by changing the prior on its variables and show explicit links with other models. In Section 4, we will review two approaches that characterize the phase diagram of the RBM and in particular its compositional phase, based on two different hypotheses over the structure of the parameters of the model. Finally, in Section 5, we will show some theoretical development helping to understand the formation of patterns inside the machine and how we can use the mean-field or TAP equations to learn the model.

## 2. Definition of the model and learning equations

### 2.1. Definition of the RBM

The RBM is an Ising model (or equivalently, a Markov random field), defined on a bipartite graph structure over two layers of variables: the visible nodes $s_i$, for $i = 1,\dots,N_{\rm v}$ and the hidden nodes $\tau_a = 1,\dots,N_{\rm h}$, with $N_{\rm v}$ and $N_{\rm h}$ denoting the numbers of visible and hidden nodes, respectively. In the following, we will use $i,j,k,\dots$ to enumerate the visible variables and $a,b,c,\dots$ for the hidden ones. No connection between any pair of visible or hidden nodes occurs . Hence, we will call $\boldsymbol{w}$ the coupling or weight matrix and denote its elements as $w_{ia}$ since no other interactions are present (such as $w_{ij}$ or $w_{ab}$). In addition to the pairwise coupling matrix $\boldsymbol{w}$, each visible and hidden node can have a local magnetic field, or local bias (we will refer to it as bias in the rest of the article), respectively named $\theta_i$ and $\eta_a$. We can introduce the following Hamiltonian:

$$\mathcal{H}[\boldsymbol{s},\boldsymbol{\tau}] = -\sum_{ia} s_i w_{ia} \tau_a - \sum_i \theta_i s_i - \sum_a \eta_a \tau_a, \tag{1}$$

from which we define a Boltzmann distribution

$$p(\boldsymbol{s},\boldsymbol{\tau}) = \frac{1}{Z}\exp(-\mathcal{H}[\boldsymbol{s},\boldsymbol{\tau}]),$$

where $Z$ is given by

$$Z = \sum_{\{\boldsymbol{s}\},\{\boldsymbol{\tau}\}} \exp(-\mathcal{H}[\boldsymbol{s},\boldsymbol{\tau}]).$$

The structure of the RBM is presented in Fig. 1 where the visible nodes are represented by black dots, the hidden nodes by red dots, and the weight matrix by blue dotted lines.

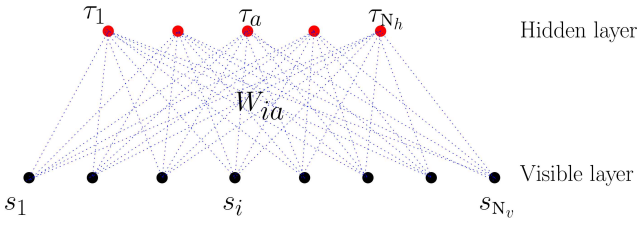



**Fig. 1.** Bipartite structure of the RBM.

The benefit of having a bipartite structure is that, when keeping fixed an entire layer, in our case all the visible or all the hidden nodes, the variables of the other layer become statistically independent. In other words, the measure $p(\boldsymbol{s}|\boldsymbol{\tau})$ and $p(\boldsymbol{\tau}|\boldsymbol{s})$ factorize over the visible/hidden nodes, respectively. This is an important property to keep in mind since

it will be used in the learning procedure of the model. We will see that this property is widely used during the learning in order to draw new samples using a Monte–Carlo Markov chain (MCMC) by Gibbs sampling.

Historically, the RBM was first defined with binary $\{0,1\}$ variables for both the visible and the hidden nodes in line with the sigmoid activation function of the perceptron, hence being directly intepretable as spin-glass model of statistical mechanics. A more general definition is considered here by introducing a prior distribution function for both the visible and hidden variables, allowing us to consider discrete or continuous variables. This generalization will allow us to see the links between RBMs and other well-known models of machine learning. From now on we will write all the equations for the generic case using the notations $q_{\rm v}(\sigma)$ and $q_{\rm h}(\tau)$ to indicate an arbitrary choice of "prior" distribution. Averaging over the RBM measure corresponding to Hamiltonian (1) will then be denoted by

$$\langle f(\boldsymbol{s},\boldsymbol{\tau})\rangle_{\mathcal{H}} = \sum_{\{s,\tau\}} p(\boldsymbol{s},\boldsymbol{\tau}) f(\boldsymbol{s},\boldsymbol{\tau}), \tag{2}$$

where $\Sigma$ can represent both discrete sums or integrals and with the RBM distribution defined from now on as

$$p(\boldsymbol{s},\boldsymbol{\tau}) = \frac{1}{Z} q_{\rm v}(\boldsymbol{s}) q_{\rm h}(\boldsymbol{\tau}) \exp(-\mathcal{H}[\boldsymbol{s},\boldsymbol{\tau}]). \tag{3}$$

It is worth mentioning that, the choice of the prior distribution can be rephrased in terms of an activation function on the conditioned distribution over the visible or hidden variables. Therefore, when specifying a prior distribution, we will systematically indicate the corresponding activation function for the hidden layer, that is $p(\boldsymbol{\tau}|\boldsymbol{s})$, which is obtained using the Bayes theorem

$$p(\boldsymbol{\tau}|\boldsymbol{s}) = \frac{p(\boldsymbol{s},\boldsymbol{\tau})}{\sum_{\tau} p(\boldsymbol{s},\boldsymbol{\tau})} = \frac{q_{\rm h}(\boldsymbol{\tau}) \exp(-\mathcal{H}[\boldsymbol{s},\boldsymbol{\tau}])}{\sum_{\{\tau\}} q_{\rm h}(\boldsymbol{\tau}) \exp(-\mathcal{H}[\boldsymbol{s},\boldsymbol{\tau}])}.$$

Before entering more into the technical details about the RBM, it is important to recall that it has been designed as a "learnable" generative model in practice. In that sense, the usual procedure is to feed the RBM with a dataset, tune its parameter $w$, $\theta$, and $\eta$ such that the equilibrium properties of the learned RBM reproduce faithfully the correlations (or the patterns) present in the dataset. In other words, it is expected that the learned model is able to produce new data statistically similar but distinct from the training set. To do so, the classical procedure is to proceed with a stochastic gradient ascent (to be explained in Subsection 2.2) of the likelihood function that can be easily expressed. Usually the learning of ML models involves the minimization of a loss function which happens here to be minus the log likelihood, thus in the following we will refer to stochastic gradient descent (SGD) instead. First, consider a set of datapoints $\{s_i^{(d)}\}$, where $d = 1,\ldots,M$ is the index of the data. The log-likelihood is given by

$$\begin{aligned}
\mathcal{L} &= \frac{1}{M}\sum_{d=1}^{M} \log\left(\sum_{\{\tau\}} p(\boldsymbol{s}^{(d)},\boldsymbol{\tau})\right) = \frac{1}{M}\sum_{d=1}^{M} \log\left(p(\boldsymbol{s}^{(d)})\right) \\
&= \frac{1}{M}\sum_{d=1}^{M}\left[\log\left(\sum_{\tau} q_{\rm v}(\boldsymbol{s}^{(d)}) q_{\rm h}(\boldsymbol{\tau}) \exp\left(-\mathcal{H}[\boldsymbol{s}^{(d)},\boldsymbol{\tau}]\right)\right)\right] - \log(Z) \\
&= \frac{1}{M}\sum_{d=1}^{M}\left[\sum_i \theta_i s_i^{(d)} + \log\left(q_{\rm v}(\boldsymbol{s}^{(d)})\right)\right. \\
&\quad \left. + \sum_a \log\left(\sum_{\tau_a} q_{\rm h}(\tau_a) \exp\left(\sum_i s_i^{(d)} w_{ia}\tau_a + \eta_a \tau_a\right)\right)\right] - \log(Z).
\end{aligned}$$

The gradient with respect to (w.r.t.) the different parameters will then take a simple form. Let us detail the computation of the gradient w.r.t. the weight matrix. By deriving the log-likelihood w.r.t. the weight matrix we get

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_{ia}} &= \frac{1}{M}\sum_{d=1}^{M} \frac{\sum_{\tau_a} q_{\rm h}(\tau_a) s_i^{(d)} \tau_a e^{\sum_i s_i^{(d)} w_{ia}\tau_a + \eta_a\tau_a}}{\sum_{\tau_a} q_{\rm h}(\tau_a) e^{\sum_i s_i^{(d)} w_{ia}\tau_a + \eta_a\tau_a}} - \langle s_i\tau_a\rangle_{\mathcal{H}} \\
&= \frac{1}{M}\sum_{d=1}^{M} s_i^{(d)} \sum_{\tau_a} \tau_a p(\tau_a|\boldsymbol{s}^{(d)}) - \langle s_i\tau_a\rangle_{\mathcal{H}} \\
&= \langle s_i\tau_a\rangle_{\rm data} - \langle s_i\tau_a\rangle_{\mathcal{H}},
\end{aligned} \tag{4}$$

where we have used the following notation:

$$\langle f(\boldsymbol{s},\boldsymbol{\tau})\rangle_{\rm data} = \frac{1}{M}\sum_{d=1}^{M}\sum_{\{\tau\}} f(\boldsymbol{s}^{(d)},\boldsymbol{\tau}) p(\boldsymbol{\tau}|\boldsymbol{s}^{(d)}). \tag{5}$$

The gradients for the biases (or magnetic fields) are

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \langle s_i\rangle_{\rm data} - \langle s_i\rangle_{\mathcal{H}}, \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial \eta_a} = \langle \tau_a\rangle_{\rm data} - \langle \eta_a\rangle_{\mathcal{H}}. \tag{7}$$

It is interesting to note that, in expression (4), the gradient is very similar to the one obtained in the traditional inverse Ising problem with the difference that in the inverse Ising problem the first term (sometimes coined "positive term") depends only on the data, while for the RBM, we have a dependence on the model (yet simple to compute). Once the gradient is computed, the parameters of the model are updated in the following way:

$$w_{ia}^{(t+1)} = w_{ia}^{(t)} + \gamma \frac{\partial \mathcal{L}}{\partial w_{ia}}\Big|_{w_{ia}^{(t)},\theta_i^{(t)},\eta_a^{(t)}}, \tag{8}$$

$$\theta_i^{(t+1)} = \theta_i^{(t)} + \gamma \frac{\partial \mathcal{L}}{\partial \theta_i}\Big|_{w_{ia}^{(t)},\theta_i^{(t)},\eta_a^{(t)}}, \tag{9}$$

$$\eta_a^{(t+1)} = \eta_a^{(t)} + \gamma \frac{\partial \mathcal{L}}{\partial \eta_a}\Big|_{w_{ia}^{(t)},\theta_i^{(t)},\eta_a^{(t)}}, \tag{10}$$

where $\gamma$ called the learning rate tunes the speed at which the parameters are updated in a given direction, the superscript $t$ being the index of iteration. A continuous limit of the learning process can be formally defined by considering $t$ real and replacing $t+1$ by $t+{\rm d}t$, $\gamma$ by $\gamma{\rm d}t$ and letting ${\rm d}t \to 0$.

The difficulty to train an RBM resides in the difficulty to compute the second term of the gradient, the so-called "negative term", which represents , in the gradient over the weight matrix, the correlation between a visible node $i$ and a hidden node $a$ under the RBM distribution. Similarly, the gradient over the biases is difficult to compute, where the negative term is given by the mean value over the visible/hidden nodes. Depending on the value of the parameters of the model (the couplings and the biases), we can either be in a phase where it is easy to sample configurations from $p(\boldsymbol{s},\boldsymbol{\tau})$, (usually called paramagnetic phase); either be (if unlucky) in a spin glass phase, where it is exponentially hard to escape from the spurious free energy minima; either be (if lucky) in a "recall" phase where the dominant states correspond to data-like configurations. But even in the latter case, it might be difficult to transit from one state to another one with random jumps if these states are separated by large energy or free energy barriers, as in the Hopfield model for instance.

### 2.2. Stochastic gradient descent

Considering the difficulty to use Eq. (4) to learn the model (the computation of the negative term scales exponentially with the system size, and Monte Caro Markov chains (MCMC) can be very slow to converge), an efficient approximative scheme name contrastive divergence[6] (CD) has been developed in order to approximate this term. First of all, the dataset is partitioned into small subsets called minibatches, and the gradient ascent is performed sequentially over all these minibatches in a random order. As a result each gradient step is performed only over a small subset of the whole dataset at a time. In order to estimate the negative term, the principle of CD is to start many Monte–Carlo chains in parallel, as many as the number of samples in a minibatch, and to use each sample of the minibatch as an initial condition for the chain. The idea being that starting from desired equilibrium configurations and making $k$ steps — the number of MC steps is coined in the method : CD-$k$— we expect to explore nearby configurations representative of the dataset when the machine is learned; if otherwise the chains flow away they will "teach" the RBM how to adjust the parameters. The interpretation of CD is that it tends to create a basin of "attraction" centered on the datapoints where nearby configurations will be attractive to these datapoint under the Gibbs dynamics. In practice, starting from a datapoint $\boldsymbol{s}^d$ a random configuration of the hidden layer is sampled; in turn given this a configuration of the visible layer is sampled and so on for $k$ steps. For this we take advantage of the bipartite structure of the model to draw a whole visible or hidden layer at once thanks to the factorization of the conditional distribution $p(\boldsymbol{s}|\boldsymbol{\tau})$ and $p(\boldsymbol{\tau}|\boldsymbol{s})$:

$$\boldsymbol{s}^d \to \boldsymbol{\tau}_0 \sim p(\boldsymbol{\tau}|\boldsymbol{s}^d) \to \boldsymbol{s}_1 \sim p(\boldsymbol{s}|\boldsymbol{\tau}_0) \to \cdots \to \boldsymbol{s}_k \sim p(\boldsymbol{s}|\boldsymbol{\tau}_{k-1}) \to \boldsymbol{\tau}_k \sim p(\boldsymbol{\tau}|\boldsymbol{s}_k), \tag{11}$$

finally $\boldsymbol{s}_k$ and $\boldsymbol{\tau}_k$ are used to estimate the negative term. It is clear that the CD-$k$ is not directly minimizing the likelihood, or equivalently the Kullback Leibler (KL) divergence between the data distribution $p_0(\boldsymbol{s})$ and the Boltzmann one $p(\boldsymbol{s})$. In reality it minimizes the KL divergence $D_{\mathrm{KL}}(p_0||p_k)$ between the data distribution $p_0$ and the distribution obtained after $k$ MC steps $p_k$ that is defined as

$$D_{\mathrm{KL}}(p_0||p_k) = \sum_{\{\boldsymbol{s}\}} p_0(\boldsymbol{s}) \log \frac{p_k(\boldsymbol{s})}{p_0(\boldsymbol{s})}$$

$$p_k(\boldsymbol{s}_k) = \sum_{\{s_0,\ldots,s_{k-1}\}} \sum_{\{\tau_0,\ldots,\tau_{k-1}\}} \left[\prod_{l=1}^{k} p(\boldsymbol{s}_l|\boldsymbol{\tau}_{l-1}) p(\boldsymbol{\tau}_{l-1}|\boldsymbol{s}_{l-1})\right] \times p_0(\boldsymbol{s}_0).$$

In Ref. [31] it is argued that this procedure is roughly equivalent to minimizing the following KL difference:

$$\mathcal{L}_{\mathrm{CD}k} = D_{\mathrm{KL}}(p_0||p) - D_{\mathrm{KL}}(p_k||p),$$

up to an extra term considered to be small without much theoretical guaranty. The major drawback of this method is that the phase space of the learned RBM is never explored since we limit ourselves to $k$ MC steps around the data configurations, therefore it can lead to estimate very poorly the probability distribution for configurations that lie "far away" from the dataset. A simple modification has been proposed to deal with this issue in Ref. [32]. The new algorithm is called persistent-CD (pCD) and consists of having again a set of parallel MC chains, but instead of using the dataset as initial condition, they are first initialized from random initial conditions and then the state of the chains is saved from one update of the parameters to the next one. In other words, the chains are initialized one time at the beginning of the learning and are then constantly updated a few MC steps further at each update of the parameters. In that case, it is no longer needed to have as many chains as the number of samples in the mini-batch even though in order to keep the statistical error comparable between the positive and the negative terms it should be of the same order. More details can be found in Ref. [32] about pCD and in Ref. [33] for a more general introduction to the learning behavior using MC. In Section 5, we will intend to understand some theoretical and numerical aspects of the RBMs learning process.

## 3. Overview of various RBM settings

Before investigating the learning behavior of RBMs, let us have a glimpse at various RBM settings and their relation to other models, by looking at common possible priors used for the visible and hidden nodes.

### 3.1. Gaussian–Gaussian RBM

The most elementary setting is the linear RBM, where both visible and hidden nodes have Gaussian priors:

$$q_{\rm v}(s_i) = \frac{1}{\sqrt{2\pi\sigma_{\rm v}^2}} \exp\left(-\frac{s_i^2}{2\sigma_{\rm v}^2}\right),$$
$$q_{\rm h}(\tau_a) = \frac{1}{\sqrt{2\pi\sigma_{\rm h}^2}} \exp\left(-\frac{\tau_a^2}{2\sigma_{\rm h}^2}\right),$$

with $\sigma_{\rm v}$ and $\sigma_{\rm h}$ are the intrinsic variances of the visible and hidden variables, respectively. After summing over hidden variables we get a multi-variate Gaussian distribution over the visible ones. If not very sophisticated, the model is yet interesting because it presents a non-trivial learning dynamics that can be written exactly.[34–37] When using Gaussian prior, the corresponding activation function $p(\boldsymbol{\tau}|\boldsymbol{s})$ is Gaussian centered on $\sigma_{\rm h}^2 \sum_i w_{ia} s_i$:

$$p(\boldsymbol{\tau}|\boldsymbol{s}) \propto \prod_a \exp\left(-\frac{\tau_a^2}{2\sigma_{\rm h}^2} + \tau_a \sum_i w_{ia} s_i\right).$$

Let us write the marginal distribution over the visible nodes $p(\boldsymbol{s})$ (we omit the hidden bias since it can be canceled by a redefinition of the visible one), starting from Eq. (3) and integrating over the hidden variables we get

$$\begin{aligned} p(\boldsymbol{s}) &= \frac{1}{Z}\prod_i \left(\mathrm{e}^{-\frac{s_i^2}{2\sigma_{\rm v}^2}+s_i\theta_i}\right) \\ &\quad \times \prod_a \left[\int \mathrm{d}\tau_a \exp\left(-\frac{\tau_a^2}{2\sigma_{\rm h}^2} + \sum_i s_i w_{ia}\tau_a\right)\right] \\ &= \frac{1}{Z}\prod_i \left(\mathrm{e}^{-\frac{s_i^2}{2\sigma_{\rm v}^2}+s_i\theta_i}\right) \prod_a \exp\left(\frac{\sigma_{\rm h}^2}{2}\sum_{ij} s_i w_{ia} w_{ja} s_j\right) \\ &= \frac{1}{Z}\exp\left(-\boldsymbol{s}^{\rm T}\left[\frac{\mathbf{1}}{2\sigma_{\rm v}^2} - \frac{\sigma_{\rm h}^2}{2}\boldsymbol{w}\boldsymbol{w}^{\rm T}\right]\boldsymbol{s} + \boldsymbol{s}^{\rm T}\boldsymbol{\theta}\right) \\ &= \frac{1}{Z}\exp\left(-\boldsymbol{s}^{\rm T}\boldsymbol{A}\boldsymbol{s} + \boldsymbol{s}^{\rm T}\boldsymbol{\theta}\right), \end{aligned} \tag{12}$$

where we define the precision matrix $\boldsymbol{A} \equiv \frac{\mathbf{1}}{2\sigma_{\rm v}^2} - \frac{\sigma_{\rm h}^2}{2}\boldsymbol{w}\boldsymbol{w}^{\rm T}$. Now we can also identify the conditions for the existence of the measure $p(\boldsymbol{s})$. We need the matrix $\boldsymbol{A}$ to be strictly positive definite, hence that the highest eigenvalue of $\boldsymbol{w}\boldsymbol{w}^{\rm T}$ remains strictly below $1/(\sigma_{\rm v}^2\sigma_{\rm h}^2)$. More interestingly, the Gaussian prior let us write in closed form the stochastic gradients (in fact we solve the deterministic equation, not the stochastic one), hence giving us some hints on the nature of the learning dynamics of non-linear RBMs, since in any case we expect a linear regime to take place at the beginning of the learning process. In the present case, we can rewrite Eq. (4) as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_{ia}} &= \frac{1}{M}\sum_d s_i^{(d)} \sigma_{\rm h}^2 \sum_j s_j^{(d)} w_{ja} - \sigma_{\rm h}^2 \langle s_i \sum_j s_j\rangle w_{ja} \\ &= \sigma_{\rm h}^2 \left(\sum_j C_{ij} w_{ja} - \sum_j \langle s_i s_j\rangle w_{ja}\right) \\ &= \sigma_{\rm h}^2 \left(\sum_j C_{ij} w_{ja} - \sum_j A_{ij}^{-1} w_{ja}\right), \end{aligned} \tag{13}$$

where $C_{ij} = \langle s_i s_j\rangle_{\rm data} = M^{-1}\sum_d s_i^{(d)} s_j^{(d)}$ is the correlation between the nodes $i$ and $j$ in the dataset, and $\boldsymbol{A}^{-1}$ the inverse of the precision matrix. At this point, for following Ref. [36], it is convenient to use the singular value decomposition (SVD) of $\boldsymbol{w}$. We note $w_{ia} = \sum_\alpha u_i^\alpha w_\alpha v_a^\alpha$ the eigen-decomposition of the rectangular weight matrix $\boldsymbol{w}$, where the matrix $\boldsymbol{u}$ and $\boldsymbol{v}$ correspond to the left (resp. right) eigenvectors of $\boldsymbol{w}$ associated to the visible (resp. hidden) variables and $w_\alpha$ the eigenvalue associated to the mode $\alpha$. As can be seen in Eq. (12), this transformation will diagonalize the interaction term of the Hamiltonian of the system. We can now make the following change of variables:

$$\hat{s}_\alpha = \sum_i u_i^\alpha s_i, \qquad \hat{\tau}_\alpha = \sum_a v_a^\alpha \tau_a,$$

under this change of variable, the Gaussian measure factorizes where $\sum_{i,j,a} s_i w_{ia} w_{ja} s_j = \sum_\alpha \hat{s}_\alpha w_\alpha^2 \hat{s}_\alpha$ and therefore

$$-\boldsymbol{s}^{\rm T}\boldsymbol{A}\boldsymbol{s} = -\frac{1}{2}\sum_\alpha \hat{s}_\alpha \frac{1-\sigma_{\rm v}^2\sigma_{\rm h}^2 w_\alpha^2}{\sigma_{\rm v}^2}\hat{s}_\alpha.$$

Writing the distribution in this new basis we obtain

$$p(\hat{\boldsymbol{s}}) \propto \prod_\alpha \exp\left(-\frac{\hat{s}_\alpha^2}{2}\frac{1-\sigma_{\rm v}^2\sigma_{\rm h}^2 w_\alpha^2}{\sigma_{\rm v}^2}\right).$$

Hence, we can obtain an exact equation for the gradient in the basis of the SVD of the weight matrix $\boldsymbol{w}$. First, we project Eq. (13) on the modes $\alpha$–$\beta$ of the SVD of $\boldsymbol{w}$

$$\begin{aligned} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}\right)_{\alpha\beta} &= \sum_{ia} u_i^\alpha \frac{\partial \mathcal{L}}{\partial w_{ia}} v_a^\beta = \sum_{ia} u_i^\alpha \left[\langle s_i\tau_a\rangle_{\rm data} - \langle s_i\tau_a\rangle_{\mathcal{H}}\right] v_a^\beta \\ &= \langle \hat{s}_\alpha \hat{\tau}_\beta\rangle_{\rm data} - \langle \hat{s}_\alpha\hat{\tau}_\beta\rangle_{\mathcal{H}}. \end{aligned}$$

Now to simplify we discard the fluctuations associated to the stochastic gradient by considering instead the full gradient and an infinitesimal learning rate such that we can consider the iteration time to be continuous and identify $\partial\mathcal{L}/\partial w_{ia} \sim \mathrm{d}w_{ia}/\mathrm{d}t$. As a result we obtain the time derivative of the matrix $\boldsymbol{w}$ decomposed over its eigenmodes

$$\begin{aligned} \left(\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t}\right)_{\alpha\beta} &= \sum_{ia} u_i^\alpha \left(\frac{\mathrm{d}}{\mathrm{d}t}\sum_\gamma u_i^\gamma w_\gamma v_a^\gamma\right) v_a^\beta \\ &= \sum_{ia\gamma} u_i^\alpha u_i^\gamma \frac{\mathrm{d}w_\gamma}{\mathrm{d}t} v_a^\gamma v_a^\beta + u_i^\alpha \frac{\mathrm{d}u_i^\gamma}{\mathrm{d}t} w_\gamma v_a^\gamma v_a^\beta \\ &\quad + u_i^\alpha u_i^\gamma w_\gamma \frac{\mathrm{d}v_a^\gamma}{\mathrm{d}t} v_a^\beta \\ &= \delta_{\alpha\beta}\frac{\mathrm{d}w_\alpha}{\mathrm{d}t} + (1-\delta_{\alpha\beta})\left(\boldsymbol{u}^\alpha \frac{\mathrm{d}\boldsymbol{u}^\beta}{\mathrm{d}t} w_\alpha + w_\beta \frac{\mathrm{d}\boldsymbol{v}^\alpha}{\mathrm{d}t}\boldsymbol{v}^\beta\right). \end{aligned}$$

This equation shows that, the gradient update of $\boldsymbol{w}$ can be decomposed when projected on the SVD basis of $\boldsymbol{w}$ into a gradient over the mode $w_\alpha$ and a rotation of the matrices $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$. Noticing first that $\langle \hat{s}_\alpha \hat{\tau}_\alpha \rangle = \sigma_{\rm h}^2 w_\alpha \langle \hat{s}_\alpha^2 \rangle$, we therefore end up with the following dynamics for the singular values $w_\alpha$:

$$\begin{aligned}\frac{{\rm d}w_\alpha}{{\rm d}t} &= \left(\frac{{\rm d}\boldsymbol{w}}{{\rm d}t}\right)_{\alpha\alpha} = \langle \hat{s}_\alpha \hat{\tau}_\alpha \rangle_{\rm data} - \langle \hat{s}_\alpha \hat{\tau}_\alpha \rangle_{\mathcal{H}} \\ &= \sigma_{\rm h}^2 w_\alpha \left( \langle \hat{s}_\alpha^2 \rangle_{\rm data} - \langle \hat{s}_\alpha^2 \rangle_{\mathcal{H}} \right) \\ &= \sigma_{\rm h}^2 w_\alpha \left( \langle \hat{s}_\alpha^2 \rangle_{\rm data} - \frac{\sigma_{\rm v}^2}{1-\sigma_{\rm v}^2\sigma_{\rm h}^2 w_\alpha^2} \right), \end{aligned} \tag{14}$$

where $\langle \hat{s}_\alpha^2 \rangle_{\rm data}$ denotes the variance of the components of the data on the mode $\alpha$

$$\langle \hat{s}_\alpha^2 \rangle_{\rm data} = \sum_{ia} u_i^\alpha \left( \frac{1}{M} \sum_{ij} s_i^{(d)} s_j^{(d)} \right) u_j^\alpha .$$

This first result tells us that when keeping the matrices $\boldsymbol{u}$ and $\boldsymbol{v}$ fixed, the SGD on the mode $w_\alpha$ will adjust the value of $w_\alpha$ such that the r.h.s matches the variance in the direction given by $\boldsymbol{u}^\alpha$, giving the following limit values:

$$w_\alpha^2 = \begin{cases} \dfrac{\langle \hat{s}_\alpha^2 \rangle_{\rm data} - \sigma_{\rm v}^2}{\sigma_{\rm v}^2 \sigma_{\rm h}^2 \langle \hat{s}_\alpha^2 \rangle_{\rm data}} & \text{if } \langle \hat{s}_\alpha^2 \rangle_{\rm data} > \sigma_{\rm v}^2, \\ 0 & \text{if } \langle \hat{s}_\alpha^2 \rangle_{\rm data} < \sigma_{\rm v}^2. \end{cases} \tag{15}$$

We remark that, if the empirical variance given by the data is smaller than the prior variance of the visible variables the corresponding mode is filtered out. The evolution of the matrices $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ can also be obtained[37] from the following expression in the present case (Actually these equations are given with a wrong sign in Ref. [37] which is corrected here):

$$\begin{aligned}\Omega_{\alpha\beta}^u &\equiv \left(\frac{{\rm d}\boldsymbol{u}^\alpha}{{\rm d}t}\right)^{\rm T} \boldsymbol{u}^\beta \\ &= -(1-\delta_{\alpha\beta})\sigma_{\rm h}^2 \left( \frac{w_\beta - w_\alpha}{w_\alpha + w_\beta} - \frac{w_\beta + w_\alpha}{w_\alpha - w_\beta} \right) \langle s_\alpha s_\beta \rangle_{\rm data}, \end{aligned} \tag{16}$$

$$\begin{aligned}\Omega_{\alpha\beta}^v &\equiv \left(\frac{{\rm d}\boldsymbol{v}^\alpha}{{\rm d}t}\right)^{\rm T} \boldsymbol{v}^\beta \\ &= -(1-\delta_{\alpha\beta})\sigma_{\rm h}^2 \left( \frac{w_\beta - w_\alpha}{w_\alpha + w_\beta} + \frac{w_\beta + w_\alpha}{w_\alpha - w_\beta} \right) \langle s_\alpha s_\beta \rangle_{\rm data} \end{aligned} \tag{17}$$

of the infinitesimal rotations of the vectors $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$. In the particular case of the Gaussian–Gaussian RBM, we can note the absence of term averaged over the model $\langle . \rangle_{\mathcal{H}}$. This is due to the fact that the SVD corresponds to the eigendecomposition of the RBM measure (that is, the Gaussian measure factorizes over the singular modes) and that Eqs. (16) and (17) involve correlation between modes $\alpha \neq \beta$ which are zero here. From Eqs. (16) and (17), we see that a steady state is found when a direction $\boldsymbol{u}^\alpha$ is found that diagonalizes the empirical covariance matrix of the dataset.

In short, the Gaussian–Gaussian RBM learns the principal components of the dataset and for each principal axis the weight matrix is adjusted until the strength of the corresponding modes $w_\alpha$ reaches the value given by Eq. (15). Of course, modes above threshold acquire a variance which matches the variance of the dataset in this direction $\langle s_\alpha^2 \rangle_{\mathcal{H}} = \langle s_\alpha^2 \rangle_{\rm data}$. We can somehow say that the Gaussian–Gaussian RBM is performing a sort of SVD of the dataset, keeping only the modes above a given threshold. It is worth noting that an analysis has been done in Ref. [35] where it is shown that updating the parameters of the model using the $k$CD approximation converges toward the same solution as the one obtained by maximizing the likelihood of the model.

We can illustrate the learning mechanism in simple cases where it is possible to solve explicitly the dynamics. First assume that the RBM has found the principal axes, i.e., consider the matrices $\boldsymbol{u}$ and $\boldsymbol{v}$ to be fixed. In this case the quantity $\langle \hat{s}_\alpha^2 \rangle_{\rm data}$ remains constant. Letting

$$x_\alpha = \sigma_{\rm v}^2 \sigma_{\rm h}^2 w_\alpha^2 \quad \text{and} \quad \delta_\alpha = \frac{\langle \hat{s}_\alpha^2 \rangle_{\rm Data} - \sigma_{\rm v}^2}{\sigma_{\rm v}^2},$$

and rescaling time as $t\sigma_{\rm v}^2\sigma_{\rm h}^2 \to t$, equation (14) then is rewritten as

$$\dot{x}_\alpha = 2x_\alpha \left( \delta_\alpha - \frac{x_\alpha}{1-x_\alpha} \right),$$

and we obtain a solution of the form

$$x_\alpha(t) = f_\alpha^{-1}(\delta_\alpha t),$$

with

$$\begin{aligned} f_\alpha(x) &= \log \frac{x}{x_\alpha(0)} - \frac{1}{1+\delta_\alpha} \log \frac{\gamma_\alpha - x}{\gamma_\alpha - x_\alpha(0)}, \\ \gamma_\alpha &= \frac{\delta_\alpha}{1+\delta_\alpha}. \end{aligned}$$

For $\delta_\alpha \ll 1$ we get a sigmoid type behavior

$$\frac{x_\alpha(t)}{x_\alpha(0)} = \frac{\delta_\alpha \, {\rm e}^{\delta_\alpha t}}{\delta_\alpha + x_\alpha(0)\left({\rm e}^{\delta_\alpha t} - 1\right)}.$$

To illustrate the rotation of the modes, consider now the situation where there are 2 modes $u_\alpha$, $\alpha = 1,2$ which are a linear combination of two dominant modes of the data $\{\hat{u}_1, \hat{u}_2\}$ with identical orientation taken in this order, all other modes are considered to be already properly aligned with the data. Let then $\theta$ represent the angle between $u_1$ and $\hat{u}_1$ (and also between $u_2$ and $\hat{u}_2$ see Fig. 2). Equation (16) for this pair of modes is then rewritten as

$$\frac{{\rm d}\theta}{{\rm d}t} = -\sigma_{\rm h}^2 \left( \frac{w_\alpha^2 + w_\beta^2}{w_\alpha^2 - w_\beta^2} \right) \langle s_1 s_2 \rangle_{\rm Data}(t),$$

with

$$\begin{aligned} \langle s_1 s_2 \rangle_{\rm Data}(t) &= \cos\theta \sin\theta \left( \langle s_2^2 \rangle_{\rm Data} - \langle s_1^2 \rangle_{\rm Data} \right), \\ \langle s_1^2 \rangle_{\rm Data}(t) &= \cos^2\theta \langle s_1^2 \rangle_{\rm Data} + \sin^2\theta \langle s_2^2 \rangle_{\rm Data}, \end{aligned}$$

$$\langle s_2^2\rangle_{\text{Data}}(t) = \sin^2\theta\langle s_1^2\rangle_{\text{Data}} + \cos^2\theta\langle s_2^2\rangle_{\text{Data}},$$

so that finally we get a dynamical system of the form

$$\dot{x}_1 = 2x_1\left(\delta_1\cos^2\theta + \delta_2\sin^2\theta - \frac{x_1}{1-x_1}\right), \tag{18}$$

$$\dot{x}_2 = 2x_2\left(\delta_1\sin^2\theta + \delta_2\cos^2\theta - \frac{x_2}{1-x_2}\right), \tag{19}$$

$$\dot{\theta} = -\frac{1}{2}(\delta_1-\delta_2)\frac{x_1+x_2}{x_1-x_2}\sin(2\theta). \tag{20}$$

Note that at fixed $x_1$ and $x_2$ the dynamics of $\theta$ corresponds to the motion of a pendulum w.r.t the variable $\theta' = 4\theta$ shown in Fig. 2.

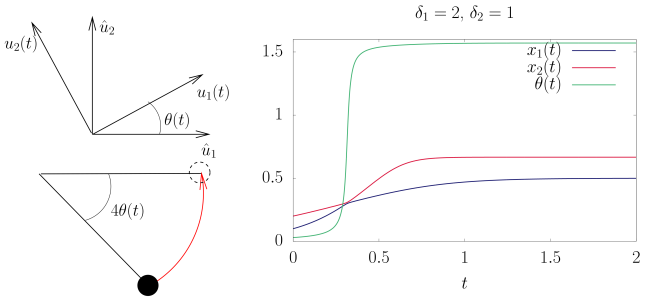



**Fig. 2.** Angle between the reference basis given by the data and the moving one given by the RBM shown on the up left panel. Equivalence with the motion of a pendulum is indicated on the left bottom panel. Solution of Eqs. (18)–(20) of two coupled modes in the linear RBM (right panel).

### 3.2. Gaussian-spherical

The Gaussian–Gaussian case is interesting as a solvable model of RBM but of limited scope, since $p(\boldsymbol{s})$ reduces in the end to a multivariate Gaussian. Next, a simple non-linear RBM which remains exactly solvable is based on the so-called spherical model.[38,39] For this model, it is possible to compute the phase diagram and the equilibrium states once the coupling matrix is given — more precisely, when the spectral density of the coupling matrix is given. Here we choose the following priors to impose a spherical constraint on the hidden nodes:

$$q_{\text{v}}(s_i) = \frac{1}{\sqrt{2\pi\sigma_{\text{v}}^2}}\exp\left(-\frac{s_i^2}{2\sigma_{\text{v}}^2}\right),$$
$$q_{\text{h}}(\boldsymbol{\tau}) = \delta\left(\sum_a \tau_a^2 - \bar{\sigma}\sqrt{N_{\text{h}}N_{\text{v}}}\right),$$

where $\bar{\sigma}$ is a parameter of the model.[40] The interest of such an RBM is first that the spherical constraint can be dealt with analytically.[40,41] Secondly the model can exhibit a phase transition unlike the Gaussian–Gaussian case. Absorbing the parameter $\sigma_{\text{v}}^2$ in the definition of the weight matrix, to follow the computation of Ref. [40], a simple analysis in the thermodynamic limit tells us that the phase transition takes place when the parameter $\bar{\sigma}$ exceeds the value $\sigma_{\text{c}}$, where $\sigma_{\text{c}}$ depends on the value of the highest mode $w_{\max}$ and of the form of the spectrum of $\boldsymbol{w}$ (typically, $\sigma_{\text{c}}^2 \propto 1/w_{\max}^2$, where the pre-factor depends on the form of the spectrum). The condensation along this mode of the visible (resp. hidden) magnetization is then given by

$$m_\alpha = \frac{1}{\sqrt{L}}\sum_i u_i^\alpha\langle s_i\rangle_{\mathcal{H}} = w_{\max}\bar{\sigma}\sqrt{\bar{\sigma}^2-\bar{\sigma}_{\text{c}}^2},$$
$$\bar{m}_\alpha^2 = \frac{1}{\sqrt{L}}\sum_a v_a^\alpha\langle\tau_a\rangle_{\mathcal{H}} = \sqrt{\bar{\sigma}^2-\bar{\sigma}_{\text{c}}^2},$$

where we have defined $L = \sqrt{N_{\text{v}}N_{\text{h}}}$. This type of RBM is again of limited scope to represent data. In the thermodynamic limit a finite number $n = \mathcal{O}(1)$ of modes can condense. They necessarily accumulate at the top of the spectrum of the weight matrix and represent a distribution concentrated on an $n$-dimensional sphere in absence of external fields while other non-condensed modes are responsible for transverse Gaussian fluctuations. The dynamical aspect of this model will be discussed in Section 5.

To end up this section let us also mention that the finite size regime is amenable to an exact analysis when restricting the weight matrix spectrum to have the property of being doubly degenerated (see Ref. [40] for details).

### 3.3. Gaussian-softmax

The case of the Gaussian mixture if rarely viewed like that, fits actually perfectly the RBM architecture. Consider here the case of Gaussian visible nodes and a set of discrete $\{0,1\}$ hidden variables with a constraint corresponding to the softmax activation function[42]

$$q_{\text{v}}(s_i) = \frac{1}{\sqrt{2\pi\sigma_{\text{v}}^2}}\exp\left(-\frac{s_i^2}{2\sigma_{\text{v}}^2}\right),$$
$$q_{\text{h}}(\boldsymbol{\tau}) = \prod_a(\delta_{\tau_a,0}+\delta_{\tau_a,1})\,\delta_{\sum_b\tau_b,1}.$$

With this formulation, we indeed see that the conditional probability of activating a hidden node is a softmax function

$$p(\tau_a = 1|\boldsymbol{s}) = \frac{\exp\left(\sum_i w_{ia}s_i+\eta_a\right)}{\sum_b\exp\left(\sum_i w_{ib}s_i+\eta_b\right)}.$$

It is easy from this expression to recognize the equations of the Gaussian mixture model (GMM),[43,44] where the latent variable $\tau_a$ indicates if a sample belongs or not to the center $a$. The position of the associated center is given by the vector $\boldsymbol{w}_a$. It is even clearer when writing the marginal over the visible nodes after summing over the hidden nodes in Eq. (3)

$$\begin{aligned} p(\boldsymbol{s}) &= \frac{1}{Z}\sum_a\exp\Big(\eta_a + \sum_i -\frac{s_i^2}{2\sigma_{\text{v}}^2} + \theta_i s_i + s_i w_{ia}\Big) \\ &= \frac{1}{Z}\sum_a\exp\Big(\eta_a + \sum_i -\frac{1}{2\sigma_{\text{v}}^2}\left(s_i - \sigma_{\text{v}}^2[w_{ia}+\theta_i]^2\right)^2 \\ &\quad + \frac{1}{2}\sigma_{\text{v}}^2[w_{ia}+\theta_i]\Big) \\ &= \frac{1}{Z'}\sum_a\rho_a\exp\Big(\sum_i -\frac{1}{2\sigma_{\text{v}}^2}\left(s_i - \sigma_{\text{v}}^2[w_{ia}+\theta_i]^2\right)^2\Big) \end{aligned}$$

by identifying

$$\rho_a \equiv \frac{\exp\left(\eta_a + \sum_i \left(\sigma_{\rm v}^2 w_{ia} + \sigma_{\rm v}^2 \theta_i\right)^2\right)}{\sum_b \exp\left(\eta_b + \sum_i \left(\sigma_{\rm v}^2 w_{ib} + \sigma_{\rm v}^2 \theta_i\right)^2\right)}, \tag{21}$$

the weight of the mode $a$ in the Gaussian mixture centered in $\boldsymbol{w}_a$. Now we can see that the extra parameter $\theta_i$ can be absorbed in the definition of the weight matrix $w'_{ia} = w_{ia} + \theta_i$. It turns out that the positive term of the gradient in Eq. (4) (ignoring $\rho_a$) corresponds to the gradient that is obtained in the GMM. This can be reformulated into the expectation maximization (EM) update by considering that $p(\tau_a|\boldsymbol{s})$ does not depend on $w_{ia}$, hence doing the "expectation" step

$$\langle s_i \tau_a \rangle_{\rm data} = \frac{1}{M} \sum_d \left(s_i^{(d)} - \sigma_{\rm v}^2 w_{ia}\right) p(\tau_a|\boldsymbol{s}^{(d)}). \tag{22}$$

If we impose that the gradient is zero, doing now the "maximization" step, we obtain

$$w_{ia}^{(t+1)} = \frac{\sum_d s_i^{(d)} p(\tau_a|\boldsymbol{s}^{(d)})}{\sigma_{\rm v}^2 \sum_d p(\tau_a|\boldsymbol{s}^{(d)})}, \tag{23}$$

where the l.h.s. is to be understood as the new values for the parameters $w_{ia}$ while the conditional distribution on the r.h.s. depends on $w_{ia}^{(t)}$. For an RBM, one would also compute the negative term of the gradient, involving the derivative of $\rho_a$ w.r.t. $w_{ia}$. We obtain the negative term

$$\langle s_i \tau_a \rangle_{\mathcal{H}} = \frac{1}{M} \sum_d \sigma_{\rm v}^2 w_{ia} \left[p(\tau_a|\boldsymbol{s}^{(d)}) - \rho_a\right]. \tag{24}$$

Again, we can recover with Eq. (24) the EM update for the density of the Gaussian mode $a$ in the GMM, by first considering that the conditioned distribution $p(\tau_a|\boldsymbol{s}^{(d)})$ does not depend on $w_{ia}$ (expectation step) and by putting the l.h.s. to zero (maximization step). The fact that when using the RBM formalism we do not obtain directly the same EM equations as in the GMM is due to the different parametrization of the parameters. In the GMM, the density of each Gaussian is defined right from the beginning as an independent parameter while when using the RBM, the density of the Gaussian depends on other parameters such as the weight matrix $\boldsymbol{w}$.

**Phase transition in the learning process** An interesting phenomenon occurs in this model when learning position of the centers of the Gaussian while submitting the variances $\sigma_{\rm v}$ of the Gaussian to an annealing process.[45] First of all, starting from a very high variance (equivalently, very high temperature), we can convince ourselves that the learning will end up finding the center of mass of the dataset. Let us therefore consider that we centered the dataset beforehand: $\sum_d s_i^{(d)} = 0, \forall i$. Then, reducing slowly the variance of each component of the mixture, we can look for the moment at which point the degenerate solution corresponding to all the centers placed at the center of masses of the dataset becomes unstable. Linearizing the EM equations (23) around this point with $\eta_a = 0$ and $w_{ia} \approx 0 + \varepsilon_{ia}$, where the $\epsilon$ are small perturbations, we can derive the threshold where the linear perturbations get amplified. The linear stability analysis leads to the following equations for the perturbation $\varepsilon$:

$$\begin{aligned}\varepsilon_{ia}^{(t+1)} &\approx \frac{\sum_d s_i^{(d)} (1 + \sum_j s_j^{(d)} \varepsilon_{ja}^{(t)} - \frac{1}{N_{\rm h}} \sum_{jb} s_j^{(d)} \varepsilon_{jb}^{(t)})}{\sigma_{\rm v}^2 \sum_d (1 + \sum_j s_j^{(d)} \varepsilon_{ja}^{(t)} - \frac{1}{N_{\rm h}} \sum_{jb} s_j^{(d)} \varepsilon_{jb}^{(t)})} \\ &= \frac{1}{\sigma_{\rm v}^2} \sum_j c_{ij} \left(\varepsilon_{ja}^{(t)} - \frac{1}{N_{\rm h}} \sum_b \varepsilon_{jb}^{(t)}\right),\end{aligned}$$

where $c_{ij}$ is the covariance matrix of the dataset. From this expression, one sees that when the variance is higher than the largest eigenvalue $\Lambda_{\rm C}$ of $\boldsymbol{c}$, i.e. ,$\sigma_{\rm v}^2 > \Lambda_{\rm C}$, the solution $w_{ia} = 0$ is stable. Then, when $\sigma_{\rm v}^2 < \Lambda_{\rm C}$, the solution is unstable and the system starts to learn something more about the dataset besides its center of mass. It is interesting to note that this threshold is very similar to the one obtained in Eq. (15) for the Gaussian–Gaussian RBM. In this model, it is then possible to study the cascade of phase transition, occurring in a hierarchical way on structured datasets.[46,47] We stress here that, even if it is possible to project the learning equations on the SVD of the weight matrix as in the two previous analysis, it does not provide much more insight since this case cannot be solved exactly by this transformation.

It is also interesting to investigate the behavior of the exact gradient (not using EM) in the presence of a learning rate $\gamma$. When using the gradient, the update equations are given by $w_{ia}^{(t+1)} = w_{ia}^{(t)} + \gamma \Delta w_{ia}$. In that case we obtain the following equation for the linear stability:

$$\varepsilon_{ia}^{(t+1)} = (1-\gamma)\varepsilon_{ia}^{(t)} + \frac{\gamma}{\sigma_{\rm v}^2} \sum_j c_{ij} \left(\varepsilon_{ja}^{(t)} - \frac{1}{N_{\rm h}} \sum_b \varepsilon_{jb}^{(t)}\right).$$

Interestingly, the threshold does not depend on the value of $\gamma$ in that case, meaning that the instability is a generic property of the learning dynamics. The only change is the speed with which the instabilities will develop.

### 3.4. Bernoulli–Gaussian RBM

The next case is the Bernoulli–Gaussian RBM where we consider the following prior:

$$\begin{aligned} q_{\rm v}(s_i) &= \frac{1}{2}\left(\delta_{s_i,0} + \delta_{s_i,1}\right), \\ q_{\rm h}(\tau_a) &= \frac{1}{\sqrt{2\pi\sigma_{\rm h}^2}} \exp\left(-\frac{\tau_a^2}{2\sigma_{\rm h}^2}\right). \end{aligned}$$

Again, a Gaussian prior implies that the activation function is Gaussian. It is interesting to consider this version of the RBM through its relation with the Hopfield model[23] was realized in Ref. [48]. Since the hidden variables are Gaussian they can be integrated out, which leads to a simple analytical form for the marginals of the visible variables. In some recent works,

the opposite approach has been done, starting with a Hopfield model and expressing it as an RBM using the Hubbard–Stratonovitch (HS) transformation (expressing the exponential of a square as a Gaussian integral) to decouple the interactions between spins.[49,50] After integrating over the hidden nodes in Eq. (3), we end up with the following distribution:

$$p(\boldsymbol{s}) = \frac{1}{Z}\exp\left(\frac{\sigma_{\rm h}^2}{2}\sum_{ij}s_i s_j\left[\sum_a w_{ia}w_{ja}\right]\right).$$

We recognize a Hopfield model where the patterns are given by the weights $w_{ia}$ of the RBM and the effective coupling between two variables $i$ and $j$ is $J_{ij} = \sum_a w_{ia}w_{ja}$. We can also consider that the variances of the hidden nodes are related to the temperature of the model.

Some experiments have been conducted in Ref. [51] in order to compare the learning process of the Hopfield model versus the Bernoulli–Gaussian RBM on artificial data generated from an Hopfield model with dicrete patterns. It is interesting to note that, when assuming discrete patterns, the inverse procedure can be formulated in terms of an approximated Hopfield model. Thus, the inference of the pattern can be done directly using a set of TAP equations of the Hopfield model, and it has been shown that the artificial patterns were inferred exactly. When using the RBM's formulation, in the absence of information over the patterns, only the subspace covered by the patterns was retrieved with a weak overlap with the true patterns. In fact, in that case the marginal over the visible nodes is a function of $\boldsymbol{w}\boldsymbol{w}^{\rm T}$, which is invariant by rotation of the $\boldsymbol{v}$ matrix. It explains why the learned weight matrix in the RBM context does not overlap with the true patterns.

With this machine, it is also possible to impose a maximum rank in order to reduce the number of parameters needed to describe the dataset giving the possibility of a trade-off between a good description of the dataset and the number of parameters. This property has been used in Ref. [50] to find global patterns in protein foldings, using the RBM version of the Hopfield model with $q$ discrete states.

### 3.5. Gaussian–Bernoulli RBM

At this point we now focus on models where the hidden layers will have a stronger impact. The integration of the hidden layer will not end up in a simple analytical form and therefore will make it difficult to understand the effect of the features and to characterize properly the learning dynamics. We first mention the Gaussian–Bernoulli case dealing with the following priors:

$$q_{\rm v}(s_i) = \frac{1}{\sqrt{2\pi\sigma_{\rm v}^2}}\exp\left(-\frac{s_i^2}{2\sigma_{\rm v}^2}\right), \tag{25}$$

$$q_{\rm h}(\tau_a) = \frac{1}{2}\left(\delta_{\tau_a,0} + \delta_{\tau_a,1}\right). \tag{26}$$

When using the discrete $\{0,1\}$ variables, we obtain the sigmoid activation function for the hidden nodes

$$p(\tau_a = 1|\boldsymbol{s}) = \frac{1}{1+\exp(-\sum_i w_{ia}s_i + \eta_a)}.$$

With this parameterization, it is natural to interpret a hidden node $\tau$ as an active feature when $\tau = 1$ and an inactive one if $\tau = 0$. When responding to a given input through the conditional probability $p(\boldsymbol{\tau}|\boldsymbol{s})$, the machine is turning on the hidden nodes corresponding to overlapping features with the input. Therefore, the input undergoes a non-linear decomposition on the learned features. Saying it that way, it is somewhat reminiscent of the independent component analysis (ICA)[52] where a matrix $\boldsymbol{X}$ is factorized on a set of independent sources or components $\boldsymbol{y}$: $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y}$. The sources here are independent in the sense that they are independently distributed. In the context of ICA, the goal is to find the inverse of the mixing matrix in order to recover the sources from the received signal. Concerning this particular RBM, it is proven[35] that under some assumptions — (i) having the same number of visible and hidden nodes, (ii) that the signal comes from a set of independent sources, and (iii) that the variance of the visible variables is much smaller than the mean of the signal — there exists a stable solution for the learning dynamics where the learned weight matrix corresponds to the un-mixing matrix of the signal. In this regime, the RBM acts as an ICA. In other words, if the signal $\boldsymbol{s}^d$ used as an input for the RBM can be written as a mixture of sources: $\boldsymbol{s} = \boldsymbol{A}\boldsymbol{y}$, a stable solution of the learning process consists in recovering the inverse mixing matrix in the weight matrix: $\boldsymbol{w} = \boldsymbol{A}^{-1}$.

To end up with this variant of the RBM, it is interesting to note that the prior variance of the visible variables here is in principle a fixed parameter. It has been noted that when using the prior (25)–(26) the mean of the conditional distribution over the visible $p(s_i|\boldsymbol{\tau})$ is stretched by the variance $\sigma_{\rm v}$. It might be useful to remove this effect by renormalizing the weight matrix and the visible biases as in Ref. [14]: $\boldsymbol{w} \to \boldsymbol{w}/\sigma_{\rm v}^2$ and $\theta_i \to \theta_i/\sigma_{\rm v}^2$. Using this parametrization, we obtain

$$p(s_i|\boldsymbol{\tau}) \approx \mathcal{N}(\theta_i + \sum_a w_{ia}\tau_a, \sigma_{\rm v}^2),$$

where $\mathcal{N}$ represents the normal distribution. Note that it is possible to include the learning of these parameters in the likelihood ascent as in Ref. [14]. It is however important to stress here that even if appealing, the possibility to tune the variance of each visible node does not solve the problem of learning individual variances of separated clusters in a dataset. Indeed, consider the problem where the dataset is formed of many well-separated clusters with distinct variances. For a given visible node $i$, its variance computed over the whole dataset or instead over a given cluster has no reason to coincide. And the the prior variance if properly learned will only account for

the global variance of this node. This should involve a more complex setting of the RBM which we will not discuss here in order to account for individual variances of clusters in a complex dataset.

### 3.6. Bernoulli–Bernoulli RBM

The last model here is traditionally the one which is implied when speaking of RBM. In that case both the visible and hidden nodes are in $\{0,1\}$ with the following priors:

$$q_{\rm v}(s_i) = \frac{1}{2}\left(\delta_{s_i,0} + \delta_{s_i,1}\right),$$
$$q_{\rm h}(\tau_a) = \frac{1}{2}\left(\delta_{\tau_a,0} + \delta_{\tau_a,1}\right).$$

The activation functions are sigmoid functions, for both the hidden and visible nodes

$$p(s_i = 1|\boldsymbol{\tau}) = \frac{1}{1+\exp(-\sum_a w_{ia}\tau_a + \theta_i)}, \tag{27}$$
$$p(\tau_a = 1|\boldsymbol{s}) = \frac{1}{1+\exp(-\sum_i w_{ia}s_i + \eta_a)}. \tag{28}$$

In that case, the prior distribution has the advantage of not having any free parameter to be determined. In practice this model is used when dealing with a discrete dataset while the Gaussian–Benoulli is for continuous ones. This model can also be generalized to the case where the hidden nodes take more than two states, see Ref. [53] for more details on this approach.

**Rectified linear units (RELU)** Let us briefly mention how the Bernoulli prior on the hidden nodes can be linked to the RELU activation function[54] for the RBM. In a work by Teh *et al.*,[55] one important shortcoming with Bernoulli prior was highlighted. With the hidden variable in $\{0,1\}$, a given pattern can be expressed $\tau = 1$, or not $\tau = 0$. Therefore the influence of a feature is binary, either 0, either a fixed amount given by the value of $\boldsymbol{w}$: it is not possible to tune this amount as a function of how strongly a hidden node responds to a visible configuration. Of course it is possible for the machine to learn many times the same pattern, but this does not seem very efficient. A simple idea to correct this problem is to duplicate many times a hidden node, keeping the same features and bias values. Then, if the probability of turning on this hidden node is $p$, the average number of activated hidden nodes for this feature will be $Np$ giving the possibility to tune the intensity of the feature.

Generalizing this idea, it is possible to construct an infinite number of replica,[56] adjusting the bias for each of them such that in order to activate more and more neurons it is necessary that the signal $\sum_i w_i s_i$ is stronger and stronger. Let us focus for a moment on a single hidden node with a feature $w_i$ and a bias $\eta$ along with its replicas $a' = 1,\ldots,N_{\rm r}$. We denote $r = \sum_i w_i s_i + \eta$ the potential associated with this neuron given the signal $\boldsymbol{s}$. The number of activated replica will be given by

$$\frac{1}{\sqrt{N_{\rm r}}}\sum_{a'=0}^{N_{\rm r}-1} \mathrm{sig}\left(r(1-a'/\sqrt{N_{\rm r}})\right) \underset{N_{\rm r}\to\infty}{\approx} \log(1+\exp(r)), \tag{29}$$

where we have defined the sigmoid function $\mathrm{sig}(x) = (1+\exp(-x))^{-1}$. The r.h.s. of Eq. (29) is very close to the RELU activation function $\mathrm{RELU}(x) = \max(0,x)$, hence showing that having all these replicas gives a similar activation function as RELU. In practice, it is not very efficient to have a large number of sigmoids for the training algorithm. An approximation is found by using the truncated Gaussian distribution. The average number of activated replica is then given by

$$\tau_a = \max(0, r + \mathcal{N}(0,\sigma_a)), \tag{30}$$

where now $\tau_a$ is a RELU hidden node and $\sigma_a$ is the variance associated with the number of activated replicas for the hidden node $a$. Equation (30) can now be seen as an approximation of the truncated-Gaussian prior for the hidden nodes

$$q_{\rm h}(\tau_a) \propto \delta_{\tau_a>0}\ \exp\left(-\frac{\tau_a^2}{2\sigma_{\rm h}}\right). \tag{31}$$

In the following section, we will focus mainly on the Bernoulli–Bernoulli setting, its equilibrium phase diagram and its learning dynamics in the mean-field regime.

## 4. Phase diagram of the Bernoulli–Bernoulli RBM

In this section, we discuss various aspects of the phase diagram of the Bernoulli–Bernoulli RBM. In the rest of the section we will use $\{\pm 1\}$ instead of the usual $\{0,1\}$ for commodity. There are (at least) two series of works dealing with the RBM in the thermodynamic limit, each of them making different hypothesis on the statistical ensemble from which the RBM is taken. In the first one[57,58] the weight matrix is taken from a simple statistical ensemble with iid elements and possibly additional sparse constraints on the patterns as will be explained in Subsection 4.1. In the second one[36,37] it is assumes that the weight matrix contains a structured part of rank $K = \mathcal{O}(1)$ in addition to a random matrix corresponding to noise; the main results of this approach will be exposed in Subsection 4.2. Both approaches are based on the replica computation[30] of the free energy. For systems with quenched disorder, this is a classical approach (the replicas or its equivalent formulation) to find the macroscopic behavior.[57,59–62]

### 4.1. Mean-field approach, the random-RBM

This MF approach to the macroscopic behavior of the RBM is based on statistical ensembles with iid elements of the weight matrix. Here, a random ensemble for the weight matrix is defined as follows. The weight matrix will be constructed

using binary pattern: $w_{ia} = \xi_{ia}/\sqrt{N_v}$. Now, each pattern is selected to be

$$\xi_{ia} = \begin{cases} 0, & p_r \sim 1 - p_i, \\ +1, & p_r \sim p_i/2, \\ -1, & p_r \sim p_i/2. \end{cases} \tag{32}$$

Using this definition, the degree of sparsity of the system is $p = \sum_i p_i/N_v$. The term random-RBM was coined by Tubiana *et al.*,[58] but Agliari *et al.*[63,64] worked on a similar model although with a different theoretical approach. In particular, they computed the phase diagram in Ref. [57]. We start by reproducing here the argument of Agliari that was developed for the RBM with a finite number of patterns before switching to the replica computation done in Tubiana's thesis.[60]

**Parallel retrieving** The usual definition of the Hopfield model (which we recall here is analogous to a Binary–Gauss RBM, see Section 3), consists in using extensive pattern $\xi_i^a = \pm 1/\sqrt{N_v}$ for all $i = 1, \ldots, N_v$, where $a = 1, \ldots, P$, with $P$ being the number of patterns. The Hopfield model in the low storage regime, where the number of patterns is fixed or scales logarithmically with the system size, is characterized by a low temperature regime made of configurations with an extensive overlap with one of the patterns. This model can be recovered from a binary–binary RBM where the number of hidden nodes has the same scaling. Hence, having $N_h \sim \log(N_v)$, we can write exactly the partition function of the binary–binary RBM in the limit of large system size

$$\begin{aligned} Z &= \sum_{\{s\},\{\tau\}} \exp\left(\beta \sum_{i,a} s_i w_{ia} \tau_a\right) \\ &= \sum_{\{s\}} \prod_a \cosh\left(\frac{\beta}{\sqrt{N_v}} \sum_i w_{ia} s_i\right) \\ &\approx \sum_{\{s\}} \exp\left(\frac{\beta^2}{2N_v} \sum_{i,j} \sum_a w_{ia} w_{ja} s_i s_j\right) \\ &= \sum_{\{s\}} \exp\left(\frac{N_v \beta^2}{2} \sum_a m_a(s)^2\right), \end{aligned}$$

recovering the Hopfield model with a square inverse temperature, where we define the magnetization along the pattern $a$ as $m_a(s)$. In Ref. [63], the authors considered a weight dilution as in Eq. (32) applied to the above binary–binary RBM, or equivalently to a Hopfield model with a rescaled temperature. It is important to mention that it is a different procedure than diluting the network itself, see Ref. [65] for more details on the other case. Having sparse patterns allows the network to retrieve more than one pattern at a time. In particular, global minima of the free energy can have an overlap with many patterns and locally stable states can be composed of a complex mixture of patterns. We reproduce below in Fig. 3 the plot from Ref. [63] showing the overlap over three and six patterns in the (almost) zero temperature limit. We observe in the left panel that one pattern is fully retrieved when the dilution is low. Then, when increasing $p_i$, more and more patterns are retrieved together until the system enters a paramagnetic phase at high dilution.

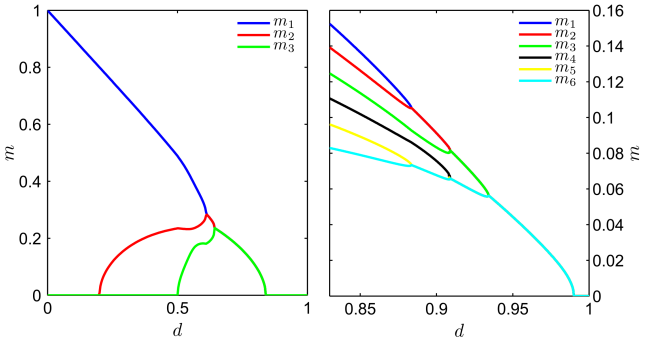



**Fig. 3.** From Ref. [63]. Overlap with different patterns when varying the dilution factor $p$ (named $d$ on the figure) at low temperature. Left: a case with 3 patterns where we can observe how at small dilution, only one pattern is fully retrieved while the second and third ones appear for larger dilution. Right: a case with 6 patterns where the figure is zoomed in the high dilution region where the branching phenomenon is occurring and all the overlaps converge toward the same value.

**Replica approach of the random-RBM** We will now follow the approach of Tubiana *et al.*[60] and give more details on the derivation. This approach is based on a Bernoulli-RELU architecture giving the possibility to have continuous positive value for the hidden variables.

The characterization of the phase diagram is based on the determination of the free energy in thermodynamic limits. Given the weight ensemble (see Eq. (32)), the weight matrix is now made of independent and sparse elements. In this context, the replica analysis can be used to perform the quenched average. The replicated interaction term can be first easily computed and gives

$$\mathbb{E}_{\boldsymbol{w}}\left[\exp\left(\beta \sum_p s_i^p w_{ia} \tau_a^p\right)\right] \approx \exp\left(\frac{p_i \beta^2}{2N} \sum_{pq} s_i^p s_i^q \tau_a^p \tau_a^q\right)$$

for the interaction term $(ia)$. The interaction between the visible and hidden nodes can be decoupled using the HS transformation

$$\begin{aligned} \exp\left(\frac{p_i^2 \beta^2}{2N} \sum_{pq} s_i^p s_i^q \tau_a^p \tau_a^q\right) &\sim \int \prod_{pq} \frac{\mathrm{d}q_{pq}\,\mathrm{d}\bar{q}_{pq}}{2\pi} \\ &\times \exp\left(-N\beta\left(q_{pq}\bar{q}_{pq} - q_{pq}\frac{p_i}{p} s_i^p s_i^q - \frac{1}{2}\beta p \tau_a^p \tau_a^q\right)\right), \end{aligned}$$

introducing the spin-glass order parameter over the replicas (we denote by $p, q, \ldots$ the replica index)

$$\begin{aligned} \bar{q}_{pq} &\sim \mathbb{E}_w[\langle \tau_a^q \tau_a^p \rangle], \\ q_{pq} &\sim \mathbb{E}_w[\langle (p_i/p) s_i^q s_i^p \rangle], \end{aligned}$$

where we note that the parameters over the visible nodes are weighted by the sparsity of the network. Using the replica symmetric ansatz, the quenched free energy can be computed and from it a set of order parameters emerges. A new order parameter is introduced in their derivation: the number $\tilde{L}$ of

hidden nodes that have a macroscopic activation $\sim \mathcal{O}(m\sqrt{N})$, while the other ones remain silent (of order 1). This parameter is reminiscent from the replica approach of the Hopfield model.[24–26] In this approach, the number of patterns that can be expressed is fixed, in order to investigate the stability of retrieving one or more patterns. The important difference here is that the sparsity $p$ imposes that the diluted weights can let many patterns be expressed at the same time. Hence, the phase diagram will be characterized by the value of the weight sparsity $p$ and the number of activated hidden nodes $\tilde{L}$. The phase diagram is computed numerically by scanning the possible value for the order parameters (see Ref. [60] for more details). It is found that when

- $p = 1$ and $\tilde{L} = 1$: no sparsity and only one hidden node is activated. At low temperature, it gives back the behavior of the well-known Hopfield model having a recall phase of the patterns. An interesting additional result when using ReLu activations is that the capacity of the network can be increased by playing with the bias on the hidden nodes, at the cost of reducing the basin of attraction of the patterns.
- $p < 1$, a ferromagnetic transition is found when imposing $\tilde{L} = 1$, where one pattern is recalled at a time.
- $p < 1$, when all the hidden nodes are all weakly activated, a SG phase is found.
- $p < 1$ and $\tilde{L}$ is such that $1 \ll \tilde{L} \ll N_{\mathrm{h}}$; a compositional phase is numerically identified. It is characterized by an intermediate number of hidden nodes strongly activated.

In this analysis, it is demonstrated that in the possible equilibrium behaviors of the random-RBM, an interesting phase mixing many patterns is present that characterizes in some way the efficient working regime of a learned RBM. It is of course a simplified case where the patterns are $\{\pm 1\}$ with a certain dilution factor. Now, the fact that there exists a family of weights where this phase exists is quite different from showing that the learning dynamics converges toward such a phase and how. In Tubiana's thesis, a stability analysis of the different phases is done showing that for a range of parameters of the RBM, the compositional phase is indeed the dominant one. Then, a certain number of numerical results are provided on the MNIST dataset which tend to confirm that the behavior of the learned RBM looks similar to a "compositional phase". It would therefore be of great interest to characterize the learning curve theoretically in order to understand how this phase is reached. It is also interesting to mention a recent work investigating the role of the diluted weights[66] during the learning in a RBM with one hidden node. In this article, it is shown that the proportion of diluted weights tends to vanish during the learning procedure. This might be a signal that when the number of hidden features is very low, the RBM automatically adjusts itself in the ferromagnetic phase described above, learning a global pattern of the dataset.

### 4.2. Mean-field approach using rank $K$ weight matrix

The difficulty with the RBM is to be able to study the phase diagram of the model without discarding the fact that during the learning, the weights $w_{ia}$ become correlated between each others: starting from independently distributed $w_{ia}$, we can observe how the spectrum of the weight matrix is modified during the learning (see Fig. 11 for instance). Classical approaches in statistical mechanics consider a set of independent weights, all identically distributed, before trying to compute the quenched free energy of the system by the replica trick (in few words, considering the quantity $Z^n$ for a given (integer) $n$, where $Z$ is the partition function, for small $n$, we can develop $Z^n \approx 1 + n\log(Z)$. The key point here is that it is generally possible to compute the quenched $Z^n$ and then make a small $n$ expansion). In the present case the hypothesis of independent weights cannot hold, as can be seen by looking at the spectrum of the weight matrix at the beginning of the learning and a few iterations later. The absorption of information by the machine prompts the development of strong correlations. This phenomenon is illustrated in Subsection 5.3 and in Fig. 11. In order to understand how these eigenvalues affect the phase diagram of the system, it is reasonable to assume a particular statistical ensemble of the weight matrix of the form

$$w_{ia} = \sum_{\alpha=1}^{K} u_i^{\alpha} w_{\alpha} v_a^{\alpha} + r_{ia}, \tag{33}$$

where $K \ll N_{\mathrm{v}}$, assuming a low-rank decomposition of the weight matrix plus some random noise $r_{ia}$. Here $\boldsymbol{r}$ is a random matrix with iid centered Gaussian elements with variance $\sigma$. With this decomposition we assume that the eigenvalues $w_{\alpha}$ correspond to some intrinsic property of a learned dataset, while the matrices $\boldsymbol{r}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ can be treated as quenched disorder and averaged over. The set of vectors $\boldsymbol{u}^{\alpha}$ and $\boldsymbol{v}^{\alpha}$ correspond approximately to the left and right eigenvectors of the matrix $\boldsymbol{w}$. We can thus start to average $Z^n$ over all these variables. Starting with the average over the random matrix $\boldsymbol{r}$, it introduces the following interaction term $\sum_{ia,p\neq q} s_i^p s_i^q \tau_a^p \tau_a^q$, where $p, q$ run over the $n$ replicas. In this term, it is possible to decouple the interaction between the visible and the hidden nodes by introducing the overlap parameters

$$Q_{pq} \sim \mathbb{E}_{r,v,u}[\langle \tau_a^q \tau_a^p \rangle], \tag{34}$$

$$\bar{Q}_{pq} \sim \mathbb{E}_{r,v,u}[\langle s_i^q s_i^p \rangle]. \tag{35}$$

Then, the form of the weight matrix, Eq. (33), leads to the following change of variable:

$$s_{\alpha} = \frac{1}{\sqrt{L}} \sum_i s_i u_i^{\alpha},$$

$$\tau_\alpha = \frac{1}{\sqrt{L}} \sum_a \tau_a v_a^\alpha,$$

where $L = \sqrt{N_\mathrm{v} N_\mathrm{h}}$. It corresponds to the projection of the visible and hidden variables over the matrices $\boldsymbol{u}$ and $\boldsymbol{v}$ coming from the SVD of $\boldsymbol{w}$. With this projection we will be able to define the order parameters of the system as the condensation of the visible and hidden nodes over the SVD modes of $\boldsymbol{w}$. Using again the Hubbard–Stratanovitch (HS) transformation in order to define the replicated magnetization

$$\exp\left(\sum_{ia} s_i^p w_{ia} \tau_a^p\right) = \exp\left(\sum_\alpha w_\alpha s_\alpha^p \tau_\alpha^p\right)$$
$$\propto \int \prod_\alpha \frac{\mathrm{d}m_\alpha^p \mathrm{d}\bar{m}_\alpha^p}{2\pi} \exp\left(-L \sum_\alpha w_\alpha (m_\alpha^p \bar{m}_\alpha^p - m_\alpha^p s_\alpha^p - \bar{m}_\alpha^p \tau_\alpha^p)\right),$$

we obtain two additional order parameters

$$m_\alpha^p \sim \mathbb{E}_{r,v,u}[\langle \tau_a^p \rangle],$$
$$\bar{m}_\alpha^p \sim \mathbb{E}_{r,v,u}[\langle s_i^p \rangle],$$

namely, the condensation of the visible (resp. hidden) nodes along the SVD modes of $\boldsymbol{w}$. After some computation, the replicated free energy is obtained as

$$\mathbb{E}_{u,v,r}[Z^n] = \int \prod_{p,\alpha} \frac{\mathrm{d}m_\alpha^p \mathrm{d}\bar{m}_\alpha^p}{2\pi} \prod_{p \neq q} \frac{\mathrm{d}Q_{pq} \mathrm{d}\bar{Q}_{pq}}{2\pi}$$
$$\times \exp\Big\{-L\Big(\sum_{p,\alpha} w_\alpha m_\alpha^p \bar{m}_\alpha^p + \frac{\sigma^2}{2} \sum_{p \neq q} Q_{pq} \bar{Q}_{pq}$$
$$- \frac{1}{\sqrt{\kappa}} A[m, Q] - \sqrt{\kappa} B[\bar{m}, \bar{Q}]\Big)\Big\}, \tag{36}$$

where $\mathbb{E}$ indicates an average over the variables that are in subscripts and $\kappa = N_\mathrm{h}/N_\mathrm{v}$. The quantities $A$ and $B$ are given by

$$A[m, Q] \equiv \log\left[\sum_{S^a \in \{-1,1\}} \mathbb{E}_u\left(\mathrm{e}^{\frac{\sqrt{\kappa}\sigma^2}{2} \sum_{p \neq q} Q_{pq} S^p S^q + \kappa^{\frac{1}{4}} \sum_{p,\alpha} (w_\alpha m_\alpha^p - \eta_\alpha) u^\alpha S^p}\right)\right], \tag{37}$$

$$B[\bar{m}, \bar{Q}] \equiv \log\left[\sum_{S^p \in \{-1,1\}} \mathbb{E}_\mathrm{v}\left(\mathrm{e}^{\frac{\sqrt{\kappa}\sigma^2}{2} \sum_{p \neq q} \bar{Q}_{pq} \tau^p \tau^q + \kappa^{-\frac{1}{4}} \sum_{p,\alpha} (w_\alpha \bar{m}_\alpha^p - \theta_\alpha) v^\alpha \tau^p}\right)\right]. \tag{38}$$

In order to avoid more cumbersome computations we will skip the details, the interested reader being referred to Ref. [36]. The phase diagram of the model is based on the behavior of the order parameters $Q$, $\bar{Q}$, $m$, and $\bar{m}$. After taking the saddle point of the free energy in the limit $L \to \infty$ keeping $\kappa$ fixed, using the replica symmetric ansatz, and letting the number of replica go to zero, it is possible to distinguish different phases according to the values of the order parameters solutions to the saddle point equations. The order parameters of the systems in the replica symmetric phase are the condensation over the SVD modes (both for the visible and hidden nodes) $\hat{m}_\alpha$ and $m_\alpha$ and the overlaps $\hat{q}$ and $q$. The saddle point equations of the free energy lead to the following self-consistent equations for the order parameters:

$$m_\alpha = \kappa^{\frac{1}{4}} \mathbb{E}_{v,x}\left[v^\alpha \tanh\left(\bar{h}(x, v)\right)\right],$$
$$q = \mathbb{E}_{v,x}\left[\tanh^2\left(\bar{h}(x, v)\right)\right], \tag{39}$$
$$\bar{m}_\alpha = \kappa^{-\frac{1}{4}} \mathbb{E}_{u,x}\left[u^\alpha \tanh\left(h(x, u)\right)\right],$$
$$\bar{q} = \mathbb{E}_{u,x}\left[\tanh^2\left(h(x, u)\right)\right], \tag{40}$$

where

$$h(x, u) = \kappa^{\frac{1}{4}}\left(\sigma\sqrt{q}x + \sum_\gamma (w_\gamma m_\gamma - \eta_\gamma) u^\gamma\right),$$
$$\bar{h}(x, v) = \kappa^{-\frac{1}{4}}\left(\sigma\sqrt{\bar{q}}x + \sum_\gamma (w_\gamma \bar{m}_\gamma - \theta_\gamma) v^\gamma\right).$$

A first look at the equations for the magnetization over the mode $\alpha$ tells us that they correspond to the usual mean-field equations of the Sherrington–Kirkpatrick model[67] projected on the SVD decomposition of the weight matrix. The same is true for the overlap, with the difference that we have an overlap parameter for each layer. Analyzing these equations, we can distinguish three phases.

- **A paramagnetic phase**, it corresponds to the case where $q = 0$, $\hat{q} = 0$, $m_\alpha = 0$, and $\hat{m}_\alpha = 0$. In the high temperature phase there exists only one minimum to the free energy.
- **A ferromagnetic phase** given by $q, \bar{q}, m_\alpha, \bar{m}_\alpha \neq 0$. In this phase, the magnetization of the system is polarized toward one or many modes $\alpha$.
- **A spin glass phase**, where $q, \bar{q} \neq 0$, but $m_\alpha = \hat{m}_\alpha = 0$. In this phase, the system is trapped into one of the many minima of the free energy that are completely uncorrelated with the SVD modes of the weight matrix.

The left panel of Fig. 4 shows the phase diagram as a function of $1/\sigma$ and of $w_\mathrm{max}/\sigma$, the ratio of the strongest mode of $\boldsymbol{w}$ to the variance $\sigma$ of the noise.

From the learning perspective, the interesting phase is the ferromagnetic one. It seems also important that the learning avoids entering into the spin glass (SG) phase. The SG phase, apart from being uncorrelated with the SVD of $\boldsymbol{w}$, can affect very badly the MCMC that is used to compute the gradient. By inspecting the phase diagram in the left panel of Fig. 4, we understand that at the beginning of the learning it is important to start with a weight matrix with a small variance $\sigma$ in order to avoid starting from the SG phase. Then, we expect during learning that one or many eigenvalues $w_\alpha$ will be

expressed and that the trajectory will drift toward the ferromagnetic phase.

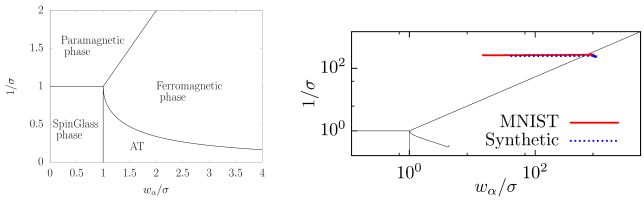



**Fig. 4.** Left: the phase diagram of the model. The $y$-axis corresponds to the variance of the noise matrix, the $x$-axis to the value of the strongest mode of $\boldsymbol{w}$. We see that the ferromagnetic phase is characterized by having strong mode eigenvalues. In this phase, the system can behave either by recalling one eigenmode of $\boldsymbol{w}$ or by composing many modes together (compositional phase). For the sake of completeness, we indicate the AT region where the replica symmetric solution is unstable, but for practical purpose we are not interested in this phase. Right: an example of a learning trajectory on the MNIST dataset (in red) and on a synthetic dataset (in blue). It shows that starting from the paramagnetic phase, the learning dynamics brings the system toward the ferromagnetic phase by learning a few strong modes.

**The nature of the ferromagnetic phase** It is instructive to look more in details at the ferromagnetic phase to understand the behavior of the RBM. We can distinguish two cases: in the first one only one eigenvalue $w_\alpha$ is learned ($w_\alpha > \sigma$) and the other ones are close to zero; in the second scenario, many eigenvalues are expressed. In fact the first case is quite simple. Since only one mode has been learned the system will condense along this mode and it will be very similar to a ferromagnet. In the second scenario, we may have many $w_\alpha$ that have been learned, i.e., which are above the noise threshold. The question then is whether the system will preferentially condense along one single mode taken out of the learned ones or whether it will be able to make compositions by condensing on several modes at the same time. In order to analyze this second scenario, it is important to recall that in order to derive the phase diagram one has to perform the quenched averaging over the matrices $\boldsymbol{u}$ and $\boldsymbol{v}$. The results will depend on the distribution that is used for the averaging. In Ref. [36], it is shown that depending on the kurtosis of the distribution taken over $\boldsymbol{u}$ and $\boldsymbol{v}$, the system can behave in different ways. Denoting with $\gamma$ the relative kurtosis (w.r.t. the normal distribution) three different behaviors are identified:

- $\gamma = 0$, e.g., the Gaussian distribution. In this case, only the strongest mode is stable, and the weaker ones are unstable w.r.t. to the strongest one. Here, the system will condense along the strongest mode only.
- $\gamma < 0$, e.g., the uniform or the Bernoulli distribution. Here the weaker modes can be metastable if they are not "too far away" from the strongest one. However the system will condense only toward one mode.
- $\gamma > 0$, e.g., a sparse Bernoulli, or the Laplace distribution. In this case, the strongest mode is unstable w.r.t. weaker ones, leaving the possibility to have condensation over many modes at the same time. This corresponds to a dual compositional phase, by reference to the terminology introduced in Ref. [58] which corresponds to combination of features instead of modes.

Hence depending on the form that will take the matrices $\boldsymbol{u}$ and $\boldsymbol{v}$ during the learning, different types of condensation may appear. This give us some insight on the way the statistical properties of the SVD of the weight matrix are reflected on the recall phase. In some cases the system might recall one macroscopic state, in another one an equilibrium state can be made of a mixture of modes. We illustrate in the right panel of Fig. 4 the learning trajectory on the phase diagram obtained both on artificial and MNIST data.

## 5. Learning RBM

Let us now discuss possible mechanisms at work during the learning of a RBM, which as we expect should have something to do with pattern formation mechanisms.[68] We start by summarizing what is understood in exactly solvable models such as the Gaussian–Gaussian and Gaussian-spherical RBMs. Then we will review a recent work[69] showing how the learning dynamics on a simple dataset for the Bernoulli–Bernoulli RBM with one hidden node can be cast into a spatial diffusion equation. Then we will investigate numerically the behavior of the RBM on the MNIST dataset. In particular, how the learned features at short time can be interpreted using the SVD of the weight matrix and how, at later time, they seem to change completely. Then leaving aside the classical approach based on the Monte–Carlo computation of the gradient — contrastive divergence,[6] persistent contrastive divergence,[32] parallel tempering[60,70,71] — we will show how to use the MF self-consistent equations in order to compute the negative term to perform the learning. Finally, we will focus on the ensemble average equations for the learning, where we show how the MF theory developed in Subsection 4.2 can be integrated numerically and lead to the learning curve of the weight matrix $\boldsymbol{w}$.

### 5.1. Learning dynamics for exactly solvable RBMs

**Gaussian–Gaussian RBM** We have already seen in Subsection 3.1 that the gradient of the Gaussian–Gaussian RBM can be computed exactly and how to characterize the growth of the eigenmodes of the weight matrix when freezing the rotation of the matrices $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$. We put additional results here, operated on an artificial dataset[37] containing 4 well-separated Gaussian clusters. Recall that the modes of the SVD of the dataset that are higher than the intrinsic variance of the visible modes $\sigma_v^2$ will be expressed, and the vectors of rotation $\boldsymbol{u}^\alpha$ will aligned themselves with the principal directions of the dataset owing to Eqs. (14), (16), and (17). We can observe

in Fig. 5 the learning curve obtained for the first eigenmodes of the system coming out of the bulk. We can also see that the first eigenvectors $\boldsymbol{u}^\alpha$, associated to the expressed eigenvalues of $\boldsymbol{w}$, are aligning with the first principal directions of the SVD of the dataset. In parallel, we see that the likelihood — that can be computed exactly here — of the system increases stepwise after each new mode is learned.

**Gaussian-spherical RBM** In the case of the Gaussian-spherical case, it is again possible to obtain an exact analytical expression for the response function of the RBM $\langle s_\alpha \tau_\beta \rangle$[40] for both the positive and negative terms, where the average is performed respectively over the dataset and the model distribution. The qualitative pictures is very similar to the previous one. As for the linear model, linear correlations between different modes vanish and therefore the matrix $\boldsymbol{u}$ has to rotate until it is properly aligned with the principal directions of the dataset. At the same time singular values get either amplified or damped. In contrary to the linear case they do not evolve independently. Instead, as seen in the left panel of Fig. 6 lower modes willing to condense exert some pressure on higher modes and accumulate at the top of the spectrum, hence pushing the whole spectrum upward. In the right panel of Fig. 6, to illustrate the result of mode condensation, we show a scatter plot containing data from the training dataset and data generated on the trained model when two modes condense.

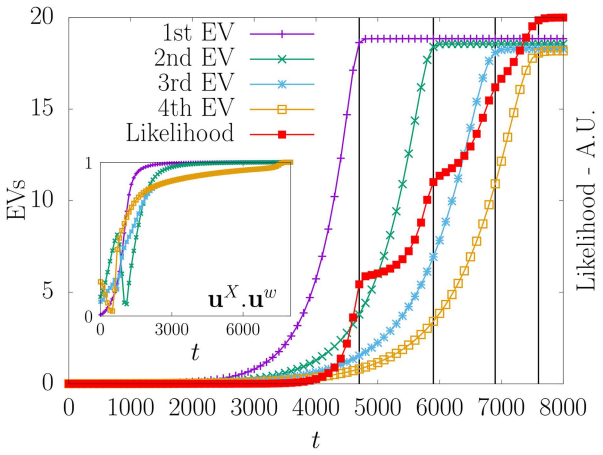



**Fig. 5.** On this artificial dataset, we observe that eigenvalues that follow $\langle s_\alpha \rangle^2 > \sigma_\mathrm{v}^2$ are learned and reach the threshold indicated by Eq. (15). In the inset, the alignment of the first four principal directions of the matrix $\boldsymbol{u}^\alpha$ of the SVD of $\boldsymbol{w}$ and of the dataset. In red, we observe that the likelihood function is increasing each time that a new mode emerges.

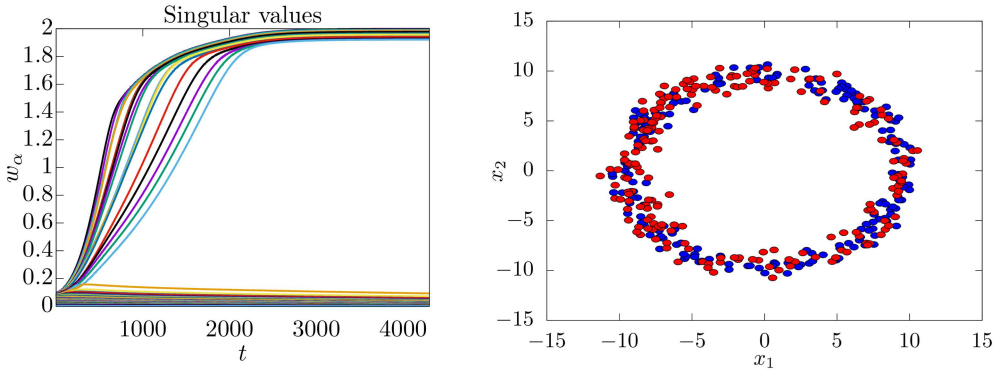



**Fig. 6.** Left: the learning curves for the modes $w_\alpha$ using an RBM with $(N_\mathrm{v}, N_\mathrm{h}) = (100, 100)$ learned on a synthetic dataset distributed in the neighborhood of a 20d ellipsoid embedded into a 100d space. Here the modes interact together: the weaker modes push the stronger ones higher, and they all accumulate at the top of the spectrum, as explained in Subsection 3.2. Right: a scatter plot projected on the first two SVD modes of the training (blue) and sampled data from the learned RBM (red) for a problem in dimension $N_\mathrm{v} = 50$ with two condensed modes. We can see that the learned matrix $\boldsymbol{u}$ captures relevant directions and that the RBM generates data perfectly similar to the one of the training set.

### 5.2. Pattern formation in the 1D Ising chain

In a recent work,[69] the formation of features is studied analytically on a RBM with one hidden node. The training dataset is generated from a 1D Ising chain with a uniform coupling constant and periodic boundary conditions. The model used for generating the data has a translational symmetry which is exploited to solve the learning dynamics exactly. There is indeed available a closed form expression for the correlation function. Thanks to the translation invariance this depends only on the relative distance between the variables. Numerically, it is found that

- the weights $\boldsymbol{w}$ function of the visible node index has a peak value for one of the visible nodes and decays with the distance to this node. Since the position of the center breaks the translation symmetry it tends to diffuse over the system during the learning.
- Using more hidden nodes (but still few), it is observed that each feature is peaked at different places and repels each other to encode the correlation patterns of the data. Again, the positions of the peaks diffuse with time even though some repulsive interaction seems to forbid them to cross. See Fig. 7 taken from Ref. [69] illustrating this phenomenon.

Now in Ref. [69], the author computed the gradient of a system with one hidden node

$$\frac{\partial \log \mathcal{L}}{\partial w_i} = \left\langle s_i \tanh\left(\beta \sum_j s_j w_j\right)\right\rangle_{\text{data}} - \tanh(w_i).$$

This expression can be developed up to the fourth order in $w$ (and $\beta$), giving in the case of the 1D Ising chain

$$\frac{\partial \log \mathcal{L}}{\partial w_i} \approx \beta(w_{i+1}+w_{i-1}) - w_i \sum_k w_k^2 + w_i^3 + \mathcal{O}(w^4, \beta w^3).$$

It is easy to identify in the first two terms the 1D discrete spatial diffusion. This equation can be cast into a spatial diffusion equation with additional term in the continuous time limit (see Ref. [69] for more details). From this small coupling expansion it is also possible to study the stationary solution in the one hidden node case and show that it is consistent with experimental results: it describes a peaked function decreasing rapidly as the distance from the center increases. An approximated weak coupling equation can also be derived in the case of two hidden units. In this case, an effective coupling between the two features vectors $w_1$ and $w_2$ is present and responsible for a repulsive interaction between the two peaks.

This illustrates nicely how the features learned by the RBM tend to describe local correlations between variables. In addition, these features diffuse over the whole system during the learning to restore the translational symmetry without crossing thanks to a repulsive interactions between them. In the next section, we will focus on the learning behavior on the MNIST dataset and see that in that case, the learned features similarly describe local correlations.

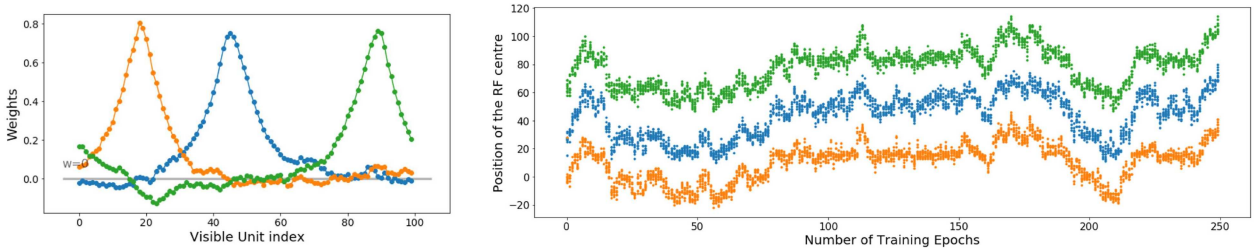



**Fig. 7.** Left: figure from Ref. [69], the value of $w_i$ for each visible site of a RBM with 3 hidden nodes trained on the dataset of the 1D homogeneous Ising model with periodic boundary condition. We see three similarly peak shaped potentials with a decreasing magnitude of similar order for the three. Each peak intends to reproduce the correlation pattern around a central node, and therefore cannot reproduce the translational symmetry of the problem. Right: figure from Ref. [69], the position of the three peaks as a function of the number of training epochs. We observe that the peaks diffuse while repelling each others. The diffusion aims at reproducing the correlation patterns of the translational symmetry, while the repelling interaction ensures that two peaks will not overlap.

### 5.3. Pattern formation in MNIST: from SVD to ICA?

The pattern formation mechanism can be studied numerically on the MNIST dataset. MNIST[8] is one of the most used real dataset in machine learning, it contains 60000 images of black and white handwritten digits of $28 \times 28$ pixels, ranging from 0 to 9. The digits are about all the same size and are at the center of the image. They are illustrated in Fig. 8.



**Fig. 8.** A subset of the MNIST dataset.

To investigate how the patterns emerge from the learning process, we inspect the features during the learning on the Bernoulli–Bernoulli RBM. The first phase of the learning can be understood thanks to a standard linear stability analysis.[36,37] For this let us recall the learning behavior of the Gaussian–Gaussian RBM analyzed in Subsection 3.1. In this simple case, the learning is triggered by the SVD of the dataset, and the growth of the modes $w_\alpha$ is controlled by how strong is the mode projected in the principal direction of the matrix $\boldsymbol{u}$. Consider now the Bernoulli–Bernoulli RBM with $\{\pm 1\}$ visible and hidden variables (to simplify), and expand the log-likelihood gradient in the limit of small $\boldsymbol{w}$ (putting the local biases to zero)

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_{ia}} &= \frac{1}{M}\sum_d s_i^{(d)} \tanh\left(\sum_j s_j^{(d)} w_{ja}\right) - \langle s_i \tau_a \rangle_{\mathcal{H}} \\ &\approx \frac{1}{M}\sum_d s_i^{(d)} \sum_j s_j^{(d)} w_{ja} - w_{ia} \\ &= \sum_j C_{ij} w_{ja} - w_{ia}.\end{aligned}$$

If we project these equations on the SVD modes of $\boldsymbol{w}$ as in Subsection 3.1, we obtain the learning dynamics

$$\mathrm{d}w_\alpha/\mathrm{d}t = w_\alpha\left[\langle \hat{s}_\alpha^2 \rangle - 1\right],$$

identical at first order in $w_\alpha$ to Eq. (14) in the Gaussian–Gaussian case, when $\sigma_\mathrm{v} = \sigma_\mathrm{h} = 1$. Hence, at the beginning of the learning, this RBM follows the same trajectory as the Gaussian–Gaussian one, where the mode $w_\alpha$ is amplified by

the principal modes of the dataset. Similarly, it can be shown that the matrix $\boldsymbol{u}$ will start to align with the principal direction of the dataset. To see how the features evolve in the non-linear regime, we train an RBM with a very low learning rate and 500 hidden nodes on MNIST. In Fig. 9 we observe as expected from the linear stability analysis, that at the beginning of the learning the first modes of the weight matrix are almost identical to the one of the SVD of the dataset. We see in particular that the features themselves correspond to modes of the dataset, meaning that the RBM starts by learning global features.

Additionally, at this stage of the learning the MC samples obtained from the RBM are typically prototypes: each sample is almost identical (or has a large overlap) with a learned feature. In fact, during the training, if we monitor samples at each epoch (keeping a low learning rate), we can see that the samples have a high overlap with one mode at the beginning, then later on with combinations of modes. To be more precise, we can distinguish different stages of the learning by inspecting the features, the produced samples, and the distance between the discretize features (taking the sign of each feature and computing the overlap). We illustrate these different stages in Fig. 10.

Finally at the end of the learning we recover localized features as in the study-case of the previous section. It has been noticed many times that these localized features are very similar to the ones given by an ICA. To which extent this aspect of learning is affected by the dataset that is considered is an open and interesting question. If we push further the learning, we observe that the RBM keeps learning more and more modes. It is not clear if the system enters into another phase (spin-glass or something else) or if it just overfits the dataset. To end up with these numerical experiments, let us look at the spectrum of $\boldsymbol{w}$, at the beginning, at an intermediate stage, and at the end of the learning. In Fig. 11, we see that starting from a Marchenko–Pastur law, coming from the spectrum of a Gaussian random matrix, quite quickly, many eigenvalues get out of the bulk as they are learned by the machine.

To summarize we have identified the following stages:

- Stage 1: at initialization, the features are completely random and therefore the histogram of distances is Gaussian and centered at zero. The spectrum of $\boldsymbol{w}$ follows the Marchenko–Pastur distribution. The RBM starts from the paramagnetic phase.

- Stage 2: the RBM enters the ferromagnetic phase, the first strongest mode of the SVD is learned by all features, giving a high positive or negative overlap in the inter-features distances while the generated samples have a high overlap with the learned features.

- Stage 3: where many modes have emerged, but the learned features remain global and close to the modes of the dataset. The histogram of distances becomes much broader but the generated sample corresponds basically to the learned features with few variety. The RBM is in a pure Mattis phase analogous to the recall phase of the Hopfield model.

- Stage 4: finally, after a much longer period, we observe that the learned features are much alike an ICA decomposition while the distances between features are still centered in zero but with a much smaller variance. Finally the generated samples look very similar to the provided dataset. The RBM is in a compositional phase, both regarding the features and the modes (the dual one).

- Stage 5: empirically, we observe that the learning of the modes of $\boldsymbol{w}$ never stops. Hence, a macroscopic number of modes is expressed and it is not clear anymore what would be the behavior of the machine in this regime, whether this corresponds to a standard spin-glass phase[62] or another unknown disordered phase.

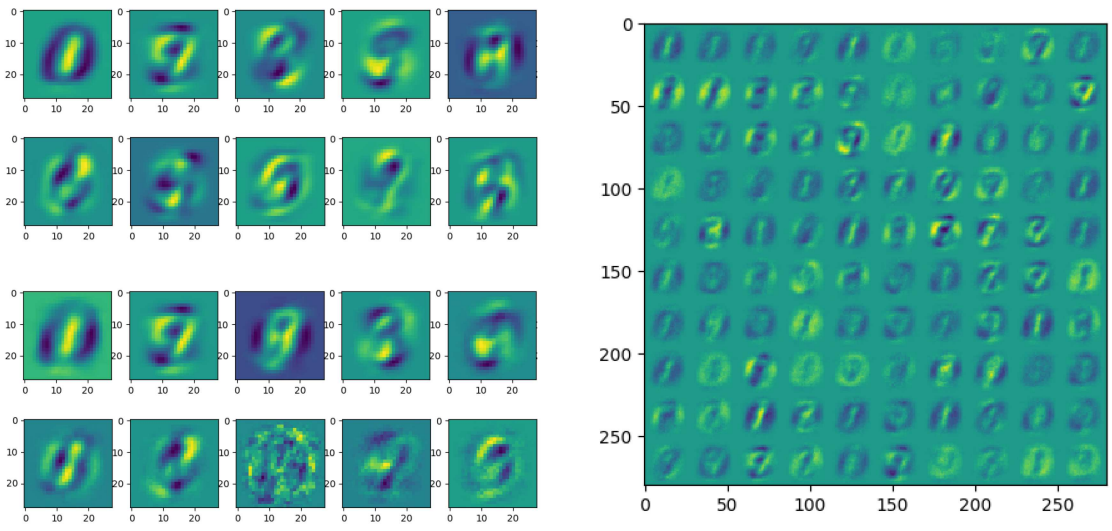


**Fig. 9.** Left: the first 10 modes of the MNIST dataset (top) and the RBM (bottom) at the beginning of the learning. The similarity between most of them is clearly visible. Right: 100 random features of the RBM at the same moment of the learning. We can see that most features correspond to a mode of the dataset when comparing with the left-top panel.

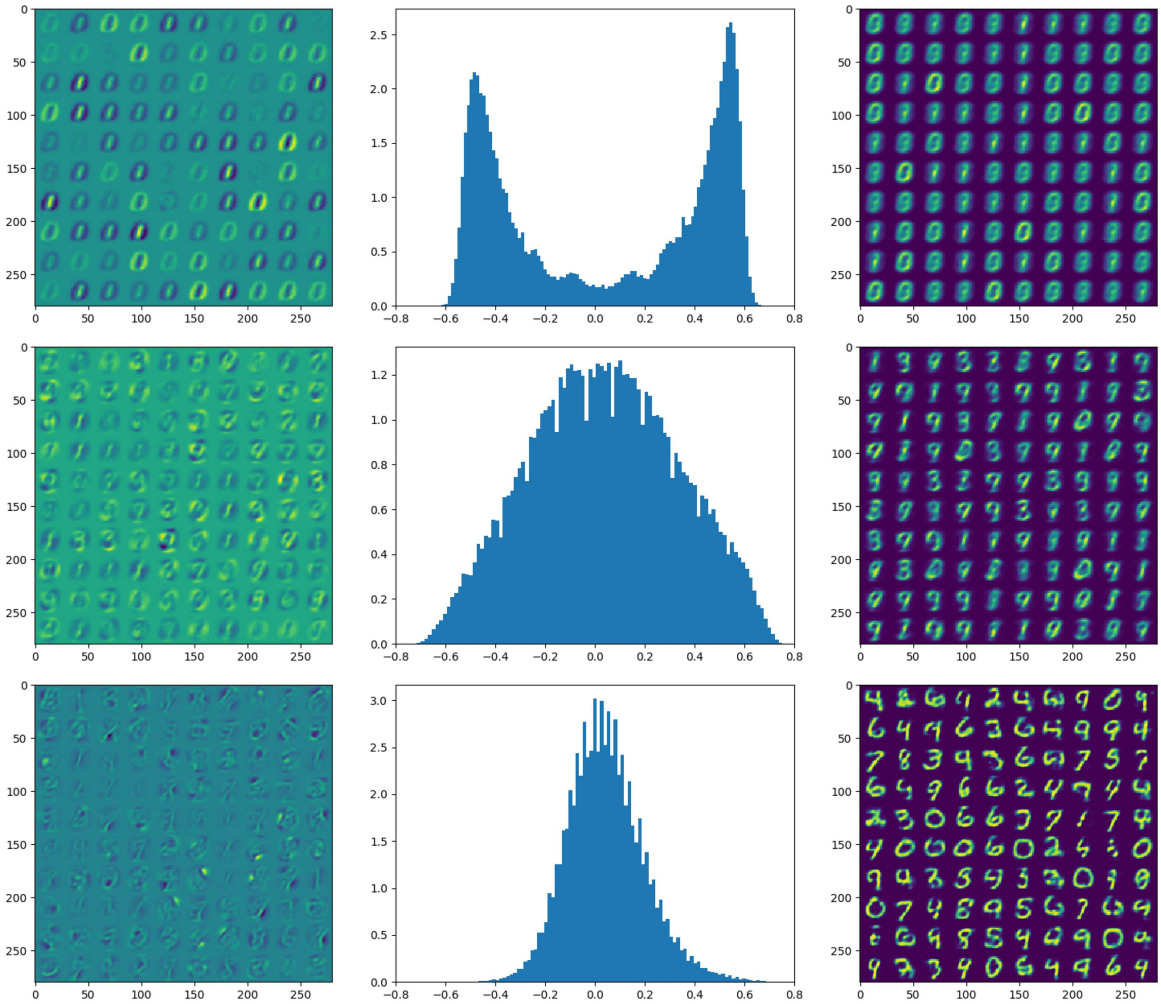


**Fig. 10.** The column represents respectively (i) the first hundred learned features, (ii) the histogram of distances between the binarized features: $W_{\pm 1} = \mathrm{sign}(\mathrm{W})$, and (iii) 100 samples generated from the learned RBM. The first row corresponds to the beginning of the learning when only one feature is learned. Looking at the histogram, we see that most of the features have a high overlap. Also, the MC samples are all similar to the learned features. On the second row, the RBM has learned many features, and therefore the histogram is wider but still centered at zero. The MC sampling however is only capable of reproducing one of the learned features. On the last row the learning is much more advanced. The features tend to be very localized and the samples correspond now to digits.

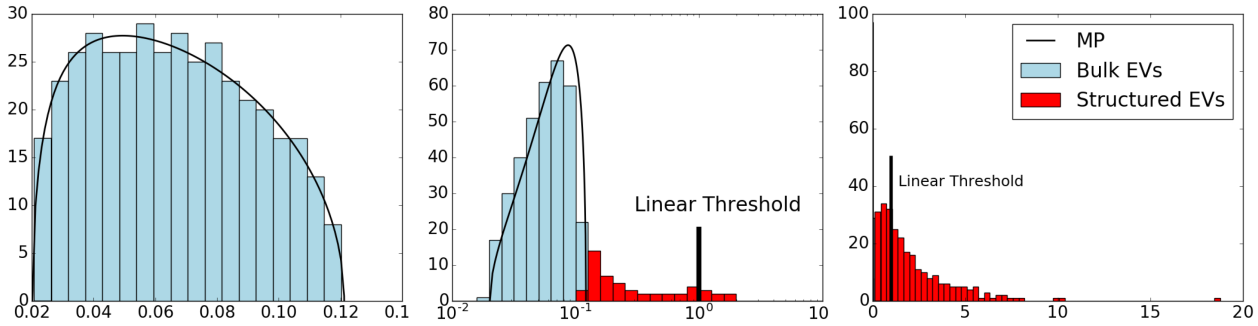



**Fig. 11.** (a) Singular values distribution of the initial random matrix compared to the Marchenko–Pastur law. (b) As the training proceeds we observe singular values passing above the threshold set by the Marchenko–Pastur law. (c) Distribution of the singular values after a long training: the Marchenko–Pastur distribution has disappeared and been replaced by a fat tailed distribution of eigenvalues mainly spreading above threshold and a peak of below-threshold singular values near zero. The distribution of eigenvalues does not get close to any standard random matrix ensemble spectrum.

In future works, it could be interesting to understand the mechanism leading to the localization of the features, in particular whether this is related to some specific tail distribution of the weight matrix spectrum. An aspect of RBMs completely absent from the previous description of the learning process is the behavior of the biases associated to the hidden nodes. These are very important since they determine the threshold above which the features are activated and their learning dynamics is quite intertwined with the modes dynamics. This aspect of the learning could be worth studying especially to

improve present learning algorithms.

### 5.4. Learning RBM using TAP equations

The difficulty of learning an RBM comes as already said from the negative term which requires to compute the thermal average of correlations between a visible and hidden nodes. In particular, when the machine starts to learn many modes, it becomes more and more difficult to estimate this term correctly using Monte–Carlo methods due to the eventually large relaxation time. In addition, to get a precise measurement, it is necessary to get many statistically independent samples in order to reduce the statistical error.

In this section we will derive the mean-field self-consistent equations that can be used to approximate the negative term by using a high-temperature expansion of the Boltzmann measure. We illustrate the method showing the result of Gabrié *et al.*[71] where an RBM has been trained by using the TAP equations. An interesting derivation using a variational approach in the case of the Gaussian–Bernoulli case has also been done in Ref. [72].

**High-temperature (Plefka) expansion** We review here a famous approach using a high-temperature expansion of the system in order to compute the mean-field magnetization. This method is both very simple to implement and also provides a way to approximate the free energy of the system in the weak couplings regime. Recent successful approaches[73,74] showed how it is possible to train an RBM using these mean-field equations. For this subsection, we will use $\{\pm 1\}$ binary variables for simplicity.

For the Ising model, it is well-known that the (naive) mean-field (nMF) approximation can be written as a set of self-consistent equations on the magnetizations, and the associated approximation of the free energy can be computed as a function of these magnetizations

$$m_i = \tanh\left(\sum_j J_{ij} m_j + h_i\right), \quad \forall i$$

$$F[\boldsymbol{m}] = \sum_i \left[\left(\frac{1-m_i}{2}\right)\log\left(\frac{1-m_i}{2}\right) + \left(\frac{1+m_i}{2}\right)\log\left(\frac{1+m_i}{2}\right)\right] - \sum_{i<j} J_{ij} m_i m_j - \sum_i m_i h_i.$$

These equations can be translated directly to the case of the RBM, with the only need to specify clearly which variables are the visible and hidden ones. One gets the following:

$$m_i = \tanh\left(\sum_a w_{ia} m_a + \theta_i\right),$$

$$m_a = \tanh\left(\sum_i w_{ia} m_i + \eta_a\right),$$

$$F[\boldsymbol{m}] = \sum_i \left[\left(\frac{1-m_i}{2}\right)\log\left(\frac{1-m_i}{2}\right) + \left(\frac{1+m_i}{2}\right)\log\left(\frac{1+m_i}{2}\right)\right] + \sum_a \left[\left(\frac{1-m_a}{2}\right)\log\left(\frac{1-m_a}{2}\right) + \left(\frac{1+m_a}{2}\right)\log\left(\frac{1+m_a}{2}\right)\right] - \sum_{i,a} w_{ia} m_i m_a - \sum_i m_i \eta_i - \sum_a m_a \theta_a.$$

Here, we remind the reader that we use the indices $i, j, k$ for the visible nodes, and $a, b, c$ for the hidden ones. We recognize in the first two lines of the free energy the entropy terms $S(m_i)$ and $S(m_a)$ of the model for respectively the visible and the hidden nodes. Note first that the nMF approximation corresponds to a first order development in $\beta$ (or in small $\boldsymbol{w}$), but it can be generalized to higher orders, recovering the so-called TAP[75] equations at the second order. Second, we can generalize this scheme to any order using the Pfleka expansion.[76,77] Let us demonstrate first how to obtain the first and second order approximations in the case of $\pm 1$ variables. To simplify the computation, we center all the terms around their mean values and make the computation for a case without local bias

$$\begin{aligned}\mathcal{H} &= -\sum_{i,a} s_i w_{ia} \tau_a \\ &= -\sum_{ia} (s_i - m_i) w_{ia} (\tau_a - m_a) - \sum_i (s_i - m_i) \sum_a w_{ia} m_a \\ &\quad - \sum_a (\tau_a - m_a) \sum_i w_{ia} m_i - \sum_{ia} m_i w_{ia} m_a.\end{aligned}$$

Using this expression, we can follow[77] and compute the magnetization in the infinite temperature limit of the following free energy:

$$-\beta A = \log\left[\sum_{\{s,\tau\}} \exp(-\beta\mathcal{H} + \sum_i \lambda_i(\beta)(s_i - m_i) + \sum_a \lambda_a(\beta)(\tau_a - m_a)\right].$$

The relation between the magnetization and the Lagrange multipliers $\lambda$ is obtained by imposing $m_i = \langle s_i \rangle_{\beta=0} = \lambda_i(0)$ and similar constraints for the hidden nodes. Then, we expand the free energy in a high temperature series

$$-\beta A = -\beta A\Big|_{\beta=0} - \beta \frac{\partial \beta A}{\partial \beta}\Big|_{\beta=0} - \frac{\beta^2}{2}\frac{\partial^2 \beta A}{\partial \beta^2}\Big|_{\beta=0} + \cdots.$$

With our Hamiltonian, we can compute the first and second orders easily

$$-\frac{\partial \beta A}{\partial \beta}\Big|_{\beta=0} = \langle \mathcal{H} \rangle = \sum_{ia} m_i w_{ia} m_a,$$

$$-\frac{\partial^2 \beta A}{\partial \beta^2}\Big|_{\beta=0} = \frac{1}{2}\sum_{ia} w_{ia}^2 (1 - m_i^2)(1 - m_a^2),$$

where we have used the following identities for the second order computation:

$$\frac{\partial^2 \beta A}{\partial \beta \partial m_i}\Big|_{\beta=0} = -\sum_a w_{ia} m_a,$$

$$\frac{\partial^2 \beta A}{\partial \beta \partial m_a}\Big|_{\beta=0} = -\sum_i w_{ia} m_i.$$

As show in Ref. [73], the expansion can be easily extended

to the third order without a big computational cost due to the particular topology of the RBM. Deriving the free energy obtained at this order w.r.t. the magnetization, we obtain the self-consistent set of equations defining the TAP equations for the RBM

$$m_i = \tanh\left(\sum_a w_{ia} m_a - \sum_a w_{ia}^2 m_i (1-m_a^2)\right), \tag{41}$$

$$m_a = \tanh\left(\sum_i w_{ia} m_a - \sum_i w_{ia}^2 m_a (1-m_i^2)\right). \tag{42}$$

Hence, a solution of the TAP equations should satisfy Eqs. (41) and (42) and give us at the same time the approximated free energy associated to this solution

$$F[\boldsymbol{m}] = \sum_i S(m_i) + \sum_a S(m_a) - \sum_{ia} w_{ia} m_i m_a + \sum_{ia} \frac{w_{ia}^2}{2} (1-m_i)^2 (1-m_a)^2. \tag{43}$$

We can now use these mean-field equations to learn the RBM. First, we need to take into account the fact that many solutions to Eqs. (41) and (42) exist, each one with a given value of the free energy. Hence, the partition function can be approximated by

$$Z = \sum_\gamma \mathrm{e}^{-F(m_i^{(\gamma)}, m_a^{(\gamma)})},$$

where the sum runs over all the possible solutions to the mean-field equations (41) and (42), weighted by the free energy given in Eq. (43). Using this approximation in the computation of the likelihood we obtain the following gradient:

$$\frac{\partial \mathcal{L}}{\partial w_{ia}} = \langle s_i \tau_a \rangle_{\text{data}} - \langle m_i m_a + w_{ia}^2 (1-m_i^2)(1-m_a^2) \rangle_{\text{MF}},$$

where

$$\langle O \rangle_{\text{MF}} = \frac{\sum_\gamma O_\gamma \mathrm{e}^{-F_\gamma}}{\sum_{\gamma'} \mathrm{e}^{-F_{\gamma'}}} \tag{44}$$

corresponds to the model average over all the solutions of the mean-field equations. We can see here a notable difference with the approach developed in Ref. [73]. In their work, Gabrié *et al.* runs the sums over all obtained fixed points from the mean-field equations divided by the number of the fixed points only. The risk is that if the mean-field equations converge toward a fixed point that is suboptimal (having a high free energy) or even spurious (being a a maximum of the free energy) the estimation of the negative term will be polluted by such fixed points. More details on the Plefka expansion on bipartite Ising model can be found in Ref. [78]. As a final remark, let us insist on the fact that, even if the convergence of the TAP equations is not guaranteed, problems of convergence are practically not met in the ferromagnetic phase. On the contrary, such problems occur quite often in the spin glass phase which we wish to avoid in the context of learning the RBM.

**Experiment with TAP learning** We show here some results obtained on MNIST using the same parameters as above but with the mean-field approximation taken from Ref. [73]. Here, the comparison is done using the persistent chain algorithm, where a set of MC chains is maintained all along the learning whenever using CD, nMF, or the TAP approximation (in the case of nMF or TAP, the chain is updated using the corresponding self-consistent equations), see Fig. 12.

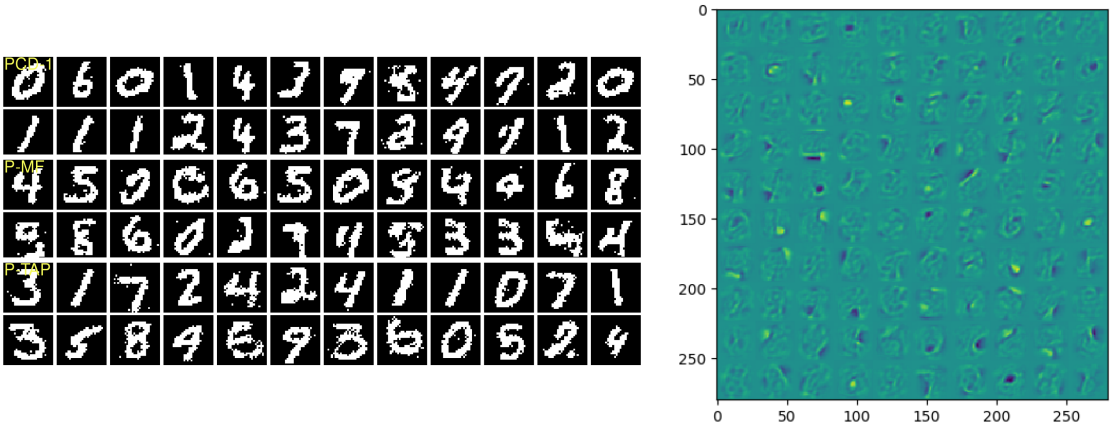



**Fig. 12.** Top: figure taken from Ref. [73], the samples taken from the permanent chain at the end of the training of the RBM. The first two lines correspond to samples generated using PCD, the second two lines to samples obtained using the P-nMF approximation, and the last two, using P-TAP. Bottom: 100 features obtained after the training, we can see that they are qualitatively very similar to the ones obtained when training the RBM with P-TAP.

First we see that the samples generated by all the three methods are qualitatively similar. Second, the features learned are also qualitatively similar to the PCD case. Therefore, on the MNIST dataset the two machines are hardly distinguished by just looking at the generated samples and learned features, indicating that the MF/TAP approximation is working very well. It is also important to point out here that, the advantage of the mean-field approximation in that case does not rely on any speed up with regard to the learning procedure. But, more importantly, it provides complementary tools such as the fixed

points as local maxima of the free energy and their associated free energy. For instance, in Ref. [79] an RBM is used as a prior distribution in the context of compressed sensing where the mean-field equations are used to infer equilibrium values of the variables. In Ref. [80], the RBM is used to reconstruct images from partial observations, again using the mean-field formulation to infer the states of the missing information.

### 5.5. Mean-field learning: ensemble average

The mean-field equations derived in Subsection 4.2 for the RBM, where the weight matrix is constructed as a low rank decomposition, can be integrated numerically in order to learn the parameters of the RBM. By contrast to the TAP equations described in Subsection5.4, which are solved on single instances, they correspond to the ensemble average (over the parameters $\boldsymbol{u}$, $\boldsymbol{v}$ and the noise), i.e., are meant to represent an average case of learning.

In the approach developed in Ref. [36], using the statistical ensemble defined in Subsection 4.2 it is possible to have a mean-field estimate of the response functions involved in the gradient of the log-likelihood. For the response term on the data we get

$$\langle s_\alpha \tau_\beta \rangle_{\text{data}} = \langle s_\alpha (s_\beta w_\beta - \theta_\beta)(1 - q_\beta[\boldsymbol{s}]) \rangle_{\text{data}},$$

where the parameter $q_\beta[\boldsymbol{s}]$ is a variant attached to mode $\beta$ of the spin-glass parameter taken as a function of $\bar{\boldsymbol{m}}$ in Eq. (39), when the visible nodes are pinned to the dataset (see Ref. [36] for details). The negative term is more complicated to compute. It depends on the fixed points obtained through Eqs. (39) and (40) for a given set of model parameters. Once the fixed points are obtained, the response terms of the RBM can be written as

$$\langle s_\alpha \tau_\beta \rangle_{\mathcal{H}} = \frac{1}{Z_{\text{MF}}} \sum_\gamma \mathrm{e}^{-Lf(m^\gamma, \bar{m}^\gamma, q^\gamma, \bar{q}^\gamma)} m_\alpha^\gamma \bar{m}_\beta^\gamma = \langle m_\alpha^\gamma \bar{m}_\beta^\gamma \rangle_{\text{MF}},$$
$$Z_{\text{MF}} = \sum_\gamma \mathrm{e}^{-Lf(m^\gamma, \bar{m}^\gamma, q^\gamma, \bar{q}^\gamma)},$$

where $\gamma$ runs over the set of fixed points, and $f$ is the mean-field free energy that can be derived from Eq. (36). These response terms allow one also to compute the skew-symmetric rotation generators of the visible and hidden singular vectors of $\boldsymbol{w}$ through

$$\Omega^u_{\alpha\beta} = \frac{w_\beta}{w_\alpha^2 - w_\beta^2} \left( \langle s_\alpha \tau_\beta \rangle_{\text{data}} - \langle s_\alpha \tau_\beta \rangle_{\mathcal{H}} \right) + \frac{w_\alpha}{w_\alpha^2 - w_\beta^2} \left( \langle s_\beta \tau_\alpha \rangle_{\text{data}} - \langle s_\beta \tau_\alpha \rangle_{\mathcal{H}} \right),$$

$$\Omega^v_{\alpha\beta} = \frac{w_\alpha}{w_\alpha^2 - w_\beta^2} \left( \langle s_\alpha \tau_\beta \rangle_{\text{data}} - \langle s_\alpha \tau_\beta \rangle_{\mathcal{H}} \right) + \frac{w_\beta}{w_\alpha^2 - w_\beta^2} \left( \langle s_\beta \tau_\alpha \rangle_{\text{data}} - \langle s_\beta \tau_\alpha \rangle_{\mathcal{H}} \right).$$

With this at hand it is therefore possible to integrate numerically the learning process of the RBM random ensemble defined by Eq. (33), hence given the typical learning trajectory. If doable in principle with any arbitrary data, this was actually tested in Ref. [36] on a simple synthetic dataset made of separated clusters. The result is shown in Fig. 13.

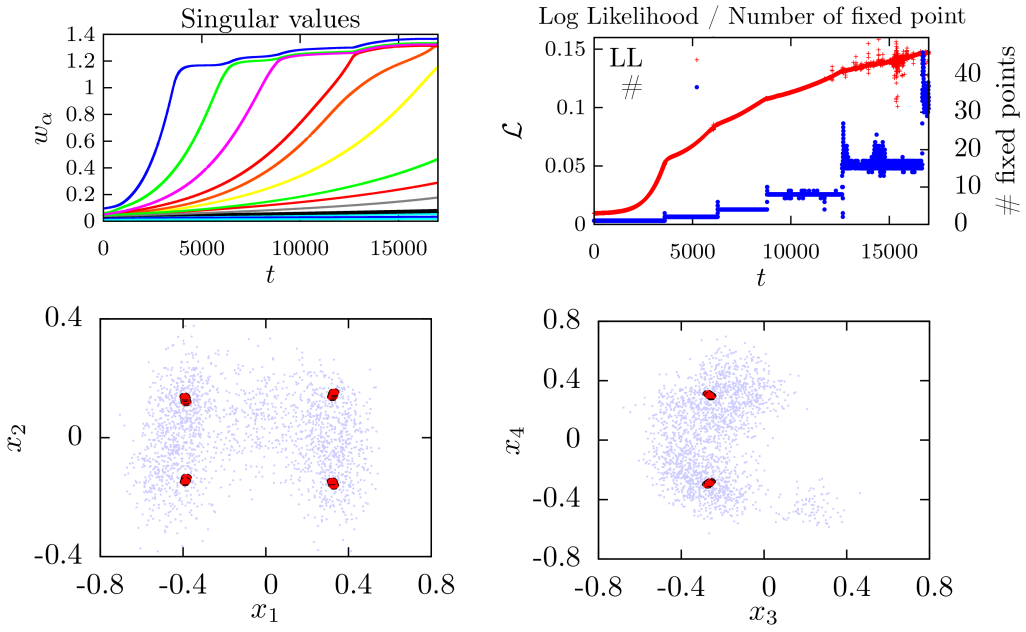



**Fig. 13.** Top panel: Results for a RBM of size $(N_{\text{v}}, N_{\text{h}}) = (1000, 500)$ learned on a synthetic dataset of $10^4$ samples having 20 clusters randomly located in a sub-manifold of dimension $d = 15$. The learning curve for the eigemodes $w_\alpha$ (left) and the associated likelihood function (right-red) together with the number of obtained fixed points at each epoch. We can see that, before the first eigenvalue is learned there is one single fixed point, then as modes are learned, the number of fixed points increases. Bottom panel: Results for an RBM of size $(N_{\text{v}}, N_{\text{h}}) = (100, 50)$ learned on a synthetic dataset of $10^4$ samples having 11 clusters randomly defined a sub-manifold of dimension $d = 5$. On the left, the scatter plot of the training data together with the position of the fixed points projected on the first two directions of the SVD of $\boldsymbol{w}$. On the right, the projection along the third and fourth axes. The results are shown after learning 5 modes, where 16 fixed points are found (in fact more than the number of hidden clusters.

We see again the different eigenvalues emerging one by one, and that each newly learned eigenvalue is triggering a jump of the likelihood together with a jump in the number of fixed points. At the end of the learning, the obtained mean-field fixed points are located at the center of each cluster of the dataset, as can be seen on the scatter plots. In Ref. [36], it is also shown that the behavior is qualitatively similar to what is routinely obtained when performing a standard learning based on PCD.

### 5.6. Other mean-field approach

Other approaches using for instance, message-passing technique such as BP have been developed in order to infer the magnetization of RBM instances. These approaches usually are correct in the limit of weak couplings, and can be used on single instance by iteratively updating a set of messages, here $\mathcal{O}(N_{\rm h}N_{\rm v})$ until convergence (see for instance Refs. [49,81]). In these works, it is shown how BP can be used to infer the magnetization or the free energy in few well-chosen cases. However, as far as practical learning tasks are concerned, it is not clear that this can be used in general when dealing with the ferromagnetic phase, as can be expected when considering structured data. In fact, it has been shown in many works that BP can have very bad convergence properties in a ferromagnetic phase when the underlying factor-graph is not a tree[82] (particularly if the couplings are strong). This would be most probably the case with RBMs. It is also worth mentioning that in the case of the inverse Ising model, BP approaches never manage to succeed because of the convergence problems[82] and the TAP solution is preferred when inferring the couplings.[83,84] However, some attempts[59,85] using BP and the replica theory on a RBM with one hidden unit were done. In that setting, it is possible to compute the marginal over the weight matrix using BP and therefore to compute its maximum likelihood given some observed datapoints. The results tend to show that, as the number of data increases, the learned features become more localized as is observed in many experiments. Managing to extend this result to the case of many hidden nodes would open the possibility to study the pattern formation using message-passing techniques. An even more recent study,[86] using a variational approach to approximate the posterior distribution of the patterns given the RBM and a dataset shows on artificial data that the patterns are recovered during the learning. Once again, the missing convincing piece in that case is the applicability to real dataset and the ability to sample complex distribution.

## 6. Conclusion

With this review, we strive at showing that not only is RBM part of a hectic field of study, but it is also an intriguing puzzle with pieces which are missing in order to be able to understand the way these models can/could assimilate complex information/more complex information. While the black box nature of the learning process starts to fade away very slowly, there are still many key aspects that we do not understand or master for such simple models. We try to list interesting leads for the future.

**Learning quality** Despite the fact that we are maximizing a likelihood function (which can not be computed) it is very hard to obtain a good indicator for comparing two learned RBMs. Even if many methods exist to compute the likelihood approximately[87,88] the obtained scores are in general not commented in regard to robust statistical analysis. If for very hard cases of image generation, it is easy to compare the results by eye inspection, there are no general method that manage to assess the quality of the samples in terms of how well the learned distribution reproduces the dataset distribution. Some recent work[89] introduced the notion of "ressemblance" and "privacy" that test the geometric repartition of the true data against the generated samples. This could be a first step defining scores according to different criteria (actually, this problem is not specific to the RBM but concerns actually most of the unsupervised learning models (GANs, VAEs, ...).

**The number of hidden nodes** It is striking that we are still unable to have a principled manner of deciding how many hidden nodes are necessary to learn datasets which are not too complex. For instance, on MNIST, it is possible to learn a machine with only 50 hidden nodes and it somehow manages to produce decent samples. The understanding on how much hidden nodes are necessary to reach a given sample quality is completely missing. In addition, the number of hidden nodes influences a lot the learning behavior of the machine, again in a way that is not fully understood.

**The landscape of free energy** When using statistical mechanics to understand RBMs, the natural question that comes in mind is about the landscape of free energy of the learned machine. It is easy to observe the mean-field fixed points obtained in the ferromagnetic phase and that they do correspond to prototypes of the dataset. Still, we do not know how these many fixed points are organized: are there low free energy paths relating them one from each others? do these paths define a network structure or instead separated clusters of low free energy?

**The landscape of learned RBMs** This is a generic question in machine learning: what is the landscape of "good" learned machines in parameter space (here the weight matrix). For supervised tasks, some consensus seems to describe a space which is globally flat where all the good models are next to one another. However this is true for deep models, in the case of RBM, apart from the permutation symmetry of the hidden nodes, we have no clue about what this landscape looks

like.

**Link between the dataset and the learned features** We have seen that in the Gaussian–Gaussian case there is a direct link between the eigen-decomposition of the dataset and the learned features. However, for the non-linear model, we do not understand how the modes of the weight matrix are linked to the dataset, nor to the associated rotation matrices.

## References


[1] Goodfellow I, Bengio Y, Courville A and Bengio Y 2016 *Deep learning*, Vol. **1** (Cambridge: MIT Press)
[2] Mehta P, Bukov M, Wang C H, Day A G R, Richardson C, Fisher C K and Schwab D J 2019 *Physics Reports* **810** 1
[3] Ronneberger O, Fischer P and Brox T 2015 *In International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241 (Springer)
[4] Carrasquilla J and Melko R G 2017 *Nat. Phys.* **13** 431
[5] Smolensky P 1986 *In Parallel Distributed Processing*, Vol. **1** (Rumelhart D and McLelland J, Ed.) pp. 194–281 (MIT Press)
[6] Hinton G E 2002 *Neural Computation* **14** 1771
[7] Ackley D H, Hinton G E and Sejnowski T J 1985 *Cognitive Science* **9** 147
[8] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 *Proc. IEEE* **86** 2278
[9] Le Roux N and Bengio Y 2008 *Neural Computation* **20** 1631
[10] Montúfar G 2016 *Restricted boltzmann machines: Introduction and review. In Information Geometry and Its Applications IV*, pp. 75–115 (Springer)
[11] Salakhutdinov R and Hinton G 2009 *Deep Boltzmann machines. In Artificial intelligence and statistics*, pp. 448–455
[12] Krizhevsky A, Hinton G, *et al.* 2009 *Learning multiple layers of features from tiny images. Technical report* (Citeseer)
[13] Yasuda M and Tanaka K 2009 *Neural Computation* **21** 3130
[14] Cho K, Ilin A and Raiko T 2011 *Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. In International conference on artificial neural networks*, pp. 10–17 (Springer)
[15] Yamashita T, Tanaka M, Yoshida E, Yamauchi Y and Fujiyoshii H 2014 *To be Bernoulli or to be Gaussian, for a restricted Boltzmann machine. In* 2014 *22nd International Conference on Pattern Recognition*, pp. 1520–1525. IEEE
[16] Hjelm R D, Calhoun V D, Salakhutdinov R, Allen E A, Adali T and Plis S M 2014 *NeuroImage* **96** 245
[17] Hu X, Huang H, Peng B, Han J, Liu N, Lv J, Guo L, Guo C and Liu T 2018 *Human brain mapping* **39** 2368
[18] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 *Generative adversarial nets. In Advances in neural information processing systems*, pp. 2672–2680
[19] Yelmen B, Decelle A, Ongaro L, Marnetto D, Tallec C, Montinaro F, Furtlehner C, Pagani L and Jay F 2021 *PLoS genetics* **17** e1009303
[20] Zhang N, Ding S F, Zhang J and Xue Y 2018 *Neurocomputing* **275** 1186
[21] KyungHyun Cho, Raiko T and Ilin A 2011 *Enhanced gradient and adaptive learning rate for training restricted Boltzmann machines*. In ICML
[22] Tang Y C and Sutskever I 2011 *Data normalization in the learning of restricted Boltzmann machines*. Department of Computer Science, University of Toronto, Technical Report UTML-TR-11-2
[23] Hopfield J J 1982 *Proc. Natl. Acad. Sci.* **79** 2554
[24] Amit D J, Gutfreund H and Sompolinsky H 1985 *Phys. Rev. A* **32** 1007
[25] Amit D J, Gutfreund H and Sompolinsky H 1985 *Phys. Rev. Lett.* **55** 1530
[26] Amit D J, Gutfreund H and Sompolinsky H 1987 *Annals of Physics* **173** 30
[27] Rosenblatt F 1958 *Psychological Review* **65** 386
[28] Gardner E 1988 *J. Phys. A: Math. Gen.* **21** 257
[29] Gardner E and Derrida B 1988 *J. Phys. A: Math. Gen.* **21** 271
[30] Mézard M, Parisi G and Virasoro M 1987 *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, Vol. 9 (World Scientific Publishing Company)
[31] Carreira-Perpinan M A and Hinton G E 2005 *On contrastive divergence learning*. In Aistats, Vol. 10, PP. 33–40. Citeseer
[32] Tieleman T 2008 *Training restricted Boltzmann machines using approximations to the likelihood gradient. In Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071
[33] Fischer A and Igel C 2014 *Pattern Recognition* **47** 25
[34] Karakida R, Okada M and Amari S I 2014 *Analyzing feature extraction by contrastive divergence learning in rbms. In Deep learning and representation learning workshop: NIPS*
[35] Karakida R, Okada M and Amari S I 2016 *Neural Networks* **79** 78
[36] Decelle A, Fissore G and Furtlehner C 2018 *J. Stat. Phys.* **172** 1576
[37] Decelle A, Fissore G and Furtlehner C 2017 *Europhys. Lett.* **119** 60001
[38] Berlin T H and Kac M 1952 *Phys. Rev.* **86** 821
[39] Stanley H E 1968 *Phys. Rev.* **176** 718
[40] Decelle A and Furtlehner C 2020 *J. Phys. A: Math. Theor.* **53** 184002
[41] Genovese G and Tantari D 2020 *J. Phys. A: Math. Theor.* **53** 094001
[42] Nijman M J and Kappen H J 1997 *International Journal of Neural Systems* **8** 301
[43] MacKay D J C and David J C 2003 *Information theory, inference and learning algorithms* (Cambridge university press)
[44] Bishop C M 2006 *Pattern recognition and machine learning*. Springer
[45] Rose K, Gurewitz E and Fox G C 1990 *Phys. Rev. Lett.* **65** 945
[46] Kloppenburg M and Tavan P 1997 *Phys. Rev. E* **55** 2089
[47] Akaho S and Kappen H J 2000 *Neural Computation* **12** 1411
[48] Barra A, Bernacchia A, Santucci E and Contucci P 2012 *Neural Networks* **34** 1
[49] Mézard M 2017 *Phys. Rev. E* **95** 022117
[50] Shimagaki K and Weigt M 2019 *Phys. Rev. E* **100** 032128
[51] Decelle A, Hwang S, Rocchi J and Tantari D 2019 arXiv:1906.11988
[52] Hyvärinen A and Oja E 2000 *Neural Networks* **13** 411
[53] Yuuki Y, Tomu K and Muneki Y 2000 *The Review of Socionetwork Strategies* **13** 253
[54] Hahnloser R H R, Sarpeshkar R, Mahowald M A, Douglas R J and Seung H S 2000 *Nature* **405** 947
[55] Teh Y W and Hinton G E 2001 *Rate-coded restricted Boltzmann machines for face recognition. In Advances in neural information processing systems*, pp. 908–914
[56] Nair V and Hinton G E 2010 *Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814
[57] Barra A, Genovese G, Sollich P and Tantari D 2018 *Phys. Rev. E* **97** 022310
[58] Tubiana J and Monasson R 2017 *Phys. Rev. Lett.* **118** 138301
[59] Huang H P 2017 *J. Stat. Mech.: Theor. Exper.* **2017** 053302
[60] Tubiana J 2018 *Restricted Boltzmann machines: from compositional representations to protein sequence analysis*. PhD thesis, ENS (Thèse de doctorat dirigée par Monasson, Rémi et Cocco, Simona Physique Paris Sciences et Lettres)
[61] Agliari E, Barra A and Tirozzi B 2019 *J. Stat. Mech.: Theor. Exper.* **2019** 033301
[62] Hartnett G S, Parker E and Geist E 2018 *Phys. Rev. E* **98** 022116
[63] Agliari E, Barra A, Galluzzi A, Guerra F and Moauro F 2012 *Phys. Rev. Lett.* **109** 268101
[64] Agliari E, Barra A, Galluzzi A and Isopi M 2014 *Neural Networks* **49** 19
[65] Wemmenhove B and Coolen A C C 2003 *J. Phys. A: Math. Gen.* **36** 9617
[66] Huang H P 2018 *J. Phys. A: Math. Theor.* **51** 08LT01
[67] Kirkpatrick S and Sherrington D 1978 *Phys. Rev. B* **17** 4384
[68] Amari S I 1977 *Biol. Cybern.* **26** 175
[69] Harsh M, Tubiana J, Cocco S and Monasson R 2020 *J. Phys. A: Math. Theor.* **53** 174002
[70] Hukushima K and Nemoto K 1996 *J. Phys. Soc. Jpn.* **65** 1604
[71] Desjardins G, Courville A, Bengio Y, Vincent P and Delalleau O 2010 *Parallel tempering for training of restricted Boltzmann machines. In Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 145–152 (Cambridge: MIT Press)
[72] Chako T and Muneki Y 2016 *J. Phys. Soc. Jpn.* **85** 034001
[73] Gabrié M, Tramel E W and Krzakala F 2015 *Training restricted Boltzmann machine via the Thouless-Anderson-Palmer free energy. In Advances in neural information processing systems*, pp. 640–648
[74] Tramel E W, Gabrié M, Manoel A, Caltagirone F and Krzakala F 2018 *Phys. Rev. X* **8** 041006

[75] Thouless D J, Anderson P W and Palmer R G 1977 *Philosophical Magazine* **35** 593
[76] Plefka T 1982 *J. Phys. A: Math. Gen.* **15** 1971
[77] Georges A and Yedidia J S 1991 *J. Phys. A: Math. Gen.* **24** 2173
[78] Maillard A, Foini L, Castellanos A L, Krzakala F, Mézard M and Zdeborová L 2019 *J. Stat. Mech.: Theor. Exp.* **2019** 113301
[79] Tramel E W, Manoel A, Caltagirone F, Gabrié M and Krzakala F 2016 *Inferring sparsity: Compressed sensing using generalized restricted Boltzmann machines. In 2016 IEEE Information Theory Workshop (ITW)*, pp. 265–269
[80] Fissore G, Decelle A, Furtlehner C and Han Y F 2019 arXiv:1912.09382
[81] Huang H P and Toyoizumi T 2015 *Phys. Rev. E* **91** 050101
[82] Lage-Castellanos A, Mulet R, Ricci-Tersenghi F and Rizzo T 2013 *J. Phys. A: Math. Theor.* **46** 135001
[83] Ricci-Tersenghi F 2012 *J. Stat. Mech.: Theor. Exp.* **2012** P08015
[84] Nguyen H C and Berg J 2012 *J. Stat. Mech.: Theor. Exp.* **2012** P03004
[85] Huang H P and Toyoizumi T 2016 *Phys. Rev. E* **94** 062310
[86] Huang H P 2020 *Phys. Rev. E* **102** 030301
[87] Salakhutdinov R and Murray I 2008 *On the quantitative analysis of deep belief networks. In Proceedings of the 25th international conference on Machine learning*, pp. 872–879
[88] Krause O, Fischer A and Igel C 2020 *Artificial Intelligence* **278** 103195
[89] Yale A, Dash S, Dutta R, Guyon I, Pavao A and Bennett K P 2020 *Generation and evaluation of privacy preserving synthetic health data* (Neurocomputing)