Restricted Boltzmann machines : from compositional representations to protein sequence analysis

Jérôme Tubiana

▶ **To cite this version:**

Jérôme Tubiana. Restricted Boltzmann machines : from compositional representations to protein sequence analysis. Physics [physics]. Université Paris sciences et lettres, 2018. English. NNT : 2018PSLEE039 . tel-02183417

HAL Id: tel-02183417
https://theses.hal.science/tel-02183417

Submitted on 15 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Restricted Boltzmann Machines : from Compositional Representations to Protein Sequence Analysis

Soutenue par
Jérôme TUBIANA
Le 29 Novembre 2018

Ecole doctorale n° 564
Physique en Île-de-France

Spécialité
Physique

Composition du jury :

Paolo DE LOS RIOS Ecole Polytechnique Fédérale de Lausanne	*Président du jury*
Riccardo, ZECCHINA Universita' Bocconi	*Rapporteur*
Lenka ZDEBOROVA CEA Saclay	*Examinatrice*
Guillaume OBOZINSKI Ecole Des Ponts et Chaussées	*Examinateur*
Simona COCCO Ecole Normale Supérieure	*Examinatrice*
Rémi MONASSON Ecole Normale Supérieure	*Directeur de thèse*

1794
ENS

CONTENTS

ABSTRACT

Restricted Boltzmann Machines (RBM) are graphical models that learn jointly a probability distribution and a representation of data. Despite their simple architecture, RBM can learn very well complex data distributions such as the handwritten digits data base MNIST. Moreover, they are empirically known to learn compositional representations of data, i.e. representations that effectively decompose configurations into their constitutive parts. However, not all variants of RBM perform equally well, and few theoretical arguments exist for these empirical observations.

In the first part of this thesis, we ask how come that such a simple model can learn such complex probability distributions and representations. By analyzing an ensemble of RBM with random weights using the replica method, we have characterized a compositional regime for RBM, and shown under which conditions (statistics of weights, choice of transfer function) it can and cannot arise. Both qualitative and quantitative predictions obtained with our theoretical analysis are in agreement with observations from RBM trained on real data.

In a second part, we present the application of RBM to protein sequence analysis and design. Owing to their large size, it is very difficult to run physical simulations of proteins, and to predict their structure and function. It is however possible to infer information about a protein structure from the way its sequence varies across organisms. For instance, Boltzmann Machines can leverage correlations of mutations to predict spatial proximity of the sequence amino-acids. Here, we have shown on several synthetic and real protein families that provided a compositional regime is enforced, RBM can go beyond structure and extract extended motifs of coevolving amino-acids that reflect phylogenic, structural and functional constraints within proteins. Moreover, RBM can be used to design new protein sequences with putative functional properties by recombining these motifs at will.

Lastly, we have designed new training algorithms and model parametrizations that significantly improve RBM generative performance, to the point where it can compete with state-of-the-art generative models such as Generative Adversarial Networks or Variational Autoencoders on medium-scale data.

RÉSUMÉ

Les Machines de Boltzmann Restreintes (Restricted Boltzmann Machines, RBM) sont des modèles graphiques capables d'apprendre simultanément une distribution de probabilité et une représentation des données. Malgré leur architecture relativement simple, les RBM peuvent reproduire très fidèlement des données complexes telles que la base de données de chiffres écrits à la main MNIST. Il a par ailleurs été montré empiriquement qu'elles peuvent produire des représentations compositionnelles des données, i.e. qui décomposent les configurations en leurs différentes parties constitutives. Cependant, toutes les variantes de ce modèle ne sont pas aussi performantes les unes que les autres, et il n'y a pas d'explication théorique justifiant ces observations empiriques.

Dans la première partie de ma thèse, nous avons cherché à comprendre comment un modèle si simple peut produire des distributions de probabilité si complexes. Pour cela, nous avons analysé un modèle simplifié de RBM à poids aléatoires à l'aide de la méthode des répliques. Nous avons pu caractériser théoriquement un régime compositionnel pour les RBM, et montré sous quelles conditions (statistique des poids, choix de la fonction de transfert) ce régime peut ou ne peut pas émerger. Les prédictions qualitatives et quantitatives de cette analyse théorique sont en accord avec les observations réalisées sur des RBM entraînées sur des données réelles.

Nous avons ensuite appliqué les RBM à l'analyse et à la conception de séquences de protéines. De part leur grande taille, il est en effet très difficile de simuler physiquement les protéines, et donc de prédire leur structure et leur fonction. Il est cependant possible d'obtenir des informations sur la structure d'une protéine en étudiant la façon dont sa séquence varie selon les organismes. Par exemple, deux sites présentant des corrélations de mutations importantes sont souvent physiquement proches sur la structure. A l'aide de modèles graphiques tels que les Machine de Boltzmann, on peut exploiter ces signaux pour prédire la proximité spatiale des acides-aminés d'une séquence. Dans le même esprit, nous avons montré sur plusieurs familles de protéines que les RBM peuvent aller au-delà de la structure, et extraire des motifs étendus d'acides-aminés en coévolution qui reflètent les contraintes phylogénétiques, structurelles et fonctionnelles des protéines. De plus, on peut utiliser les RBM pour concevoir de nouvelles séquences avec des propriétés fonctionnelles putatives par recombinaison de ces motifs.

Enfin, nous avons développé de nouveaux algorithmes d'entraînement et des nouvelles formes paramétriques qui améliorent significativement la performance générative des RBM. Ces améliorations les rendent compétitives avec l'état de l'art des modèles génératifs tels que les réseaux génératifs adversariaux ou les auto-encodeurs variationels pour des jeux de données de taille intermédiaires.

ACKNOWLEDGMENTS

Je tiens d'abord à remercier mon directeur de thèse Rémi Monasson ainsi que ma collaboratrice Simona Cocco d'avoir encadré ma thèse au cours de ces trois dernières années. J'ai bénéficié d'un environnement chaleureux et humain, d'un encadrement scientifique de très haut niveau et d'une exposition à des sujets de recherche passionnants. A ce titre, je me souviendrai de ma thèse comme de l'une de mes meilleures expériences professionnelles.

Je tiens également particulièrement à remercier Jean-François Allemand, pour ses nombreux conseils pertinents et bienveillants dont je bénéficie depuis avant mon arrivée à l'ENS en 2011. Plus généralement, cette thèse n'aurait pas été possible sans l'excellente formation scientifique dont j'ai bénéficié ici, et je souhaite donc remercier l'ensemble de l'équipe pédagogique de l'ENS. J'ai eu le plaisir de participer (à mon niveau) à la formation des étudiants de l'ENS au cours de ma thèse grâce à Frédéric Chevy, je l'en remercie.

Je remercie également Aleksandra Walczak et Lenka Zdeborova d'avoir accepté d'être respectivement ma parraine et tutrice scientifique et pour les discussions que nous avons eues. Je remercie également Guilhem Semerjian, Georges Debregeas, Didier Chatenay et Eric Aurell pour les conseils scientifiques avisés.

Je souhaite ensuite remercier chacun des membres du jury de ma soutenance de thèse: merci à Guillaume Obozinski, Lenka Zdeborova (encore), et particulièrement à Paolo de Los Rios et Riccardo Zecchina pour avoir accepté d'être rapporteurs et être venus de loin pour assister à ma soutenance de thèse.

Merci également à Viviane Sebille, Sandrine Pataccini et Laura Baron-Ledez pour leur aide administrative précieuse. Je remercie également Marc-Thierry Jaekel pour son support technique et pour avoir mis en place une infrastructure informatique solide et efficace, qui a survécu à mes innombrables benchmarking et autres essais ratés.

Au cours de ma thèse, j'ai eu le grand plaisir de rencontrer et de partager mon quotidien avec de nombreux étudiants, collègues du département de physique. Je remercie chaleureusement Ada, Aldo, Alessandro, Alexis, Arnaud, Beatriz, Clément, Dario, Elisabetta, Emmanuel, Francesca, Kevin, Lorenzo, Marco, Moshir, Sébastien, Steven, Thijs pour leur amitié, leur soutien émotionnel et scientifique et les nombreux moments chaleureux passés ensemble.

Il est difficile de réussir sa thèse sans être heureux en dehors, et au cours de ces trois années de motivations et résultats fluctuants, j'ai pu compter sur le soutien de ma famille et de mes amis. Un merci particulier à mes parents

Gérard et Joëlle, à ma tante Dany, à mon frère Rémy et ma belle-soeur Lisa pour leur soutien moral et affectif inconditionnel. Merci également à tous mes amis en dehors du laboratoire, pour leur écoute, leur amitié et leur bonne humeur contagieuse.

J'ai enfin un mot particulier pour celle qui partage ma vie. Odélia je te remercie d'avoir été à mes cotés au cours de ces années ponctuée de hauts et ses bas; je te remercie de m'avoir toujours soutenu, et tantôt écouté, consolé, aidé concrètement, ou botté le derrière lorsque c'était nécessaire ! Maintenant, à mon tour !

1

INTRODUCTION

Over the last years, deep learning [1,2], a family of machine learning algorithms based on neural networks, has dramatically improved state-of-the-art performance in numerous fields, including image [3–5] and speech recognition [6,7], natural language processing [8,9], text translation [10–12], computational medical diagnosis [13], artificial image/video generation [14,15]. These successes were notably allowed by the availability of increasingly large data sets, computational resource and software frameworks. On the other hand, our theoretical understanding of neural networks has evolved at a slower pace, and although recent theoretical developments are emerging [16–19], numerous questions remain: how can such large models with hundreds of millions of parameters not overfit the data ? Why does the non-convex optimization work so well in practice ? Why do some architectures and parameters outperform others ? Neural networks could benefit from a better theoretical understanding, as empirical knowledge can be hard to transfer from one experiment to the other. For instance, the image recognition challenge ImageNet 2015 was won using an ensemble of very deep neural networks, each consisting of 152 layers, a staggering number [5]. However, such very deep architectures should not be required to achieve human-like performance as the visual cortex is not as deep; but since the reason this model outperforms the others is unknown, we cannot reverse engineer it into a simpler architecture. Current progresses therefore essentially rely on improving optimization algorithm [20–22] and exploring increasingly more complex architectures [4,23].

Another issue raised by these successes is that as neural networks become more and more complex, they behave more and more as black-boxes whose outputs are difficult to interpret. In supervised learning, one may want to know what clues are picked up by the model to make a decision; in unsupervised learning, e.g. in probability distribution learning, one may want to know what are the characteristic features of a configuration that give it high probability. This is particularly crucial in the context of data analysis in biology, where models must be both quantitative and relatable to the underlying biological mechanisms. Owing to continuous progresses in data acquisition techniques, such as electrophysiological and fluorescence-based functional recordings of neurons in neuroscience, DNA sequencing, single RNA-sequencing and deep

mutational scans of protein fitness landscapes, the amount of available data has drastically increased. How to exploit these data in both a quantitative and easily interpretable fashion? Most often, interpretability comes at the expense of decreased quantitative performance: linear and logistic regression in supervised learning or mixture models in unsupervised learning are well understood, but rarely provide an sufficient description of data. On the other hand, deep neural networks, though powerful, may not be the best tools for the purpose of interpretation.

Statistical physics may play a key role in addressing both of these issues. Since the 80's, ideas from statistical physics have led to both fundamental and practical developments in computer science and neural networks. The physics-inspired simulated annealing optimization procedure [24] had major impact in applied computer science and engineering. Statistical physics tools were used to study learning dynamics and maximum capacity of feed-forward and recurrent neural networks such as the perceptron and the Hopfield model [25–27]. More recently, statistical physics was applied to study transitions from polynomial to non-polynomial complexity in K-satisfiability problems [28, 29], and theoretical investigation tools such as TAP and belief propagation were shown to efficiently address inference problems [30, 31]. Nowadays, connections between deep neural network optimization landscapes and spin-glass energy landscapes and between dynamics of learning and Langevin dynamics are under active investigation [16, 19]. From the perspective of data modeling, a key conceptual input from statistical physics is that very complex collective behaviors can emerge from simple interactions between individuals: the traditional example is the case of the Ising model, where long-range ferromagnetic order can arise from local couplings between spins. Conversely, this suggests that complex data may be explainable by relatively simple models. The recently developed inverse Ising procedure, which consists in finding interactions that reproduce data correlations found successes in numerous biological problems [32]. More generally, the physics top-down culture of explaining observations by minimalist models may find future applications for developing interpretable machine learning models.

My PhD, realized at the Laboratory of Theoretical Physics of ENS Paris, under the supervision of Pr. Rémi Monasson and in collaboration with Pr. Simona Cocco, at the interface between statistical physics, machine learning and bioinformatics takes place in this general context. This thesis focuses on Restricted Boltzmann Machines (RBM), a simple yet powerful generative neural network, and their application to protein sequence modeling. Though they are much simpler than deep feedforward or generative networks, they share the similar working principle of learning compositional representations of data.

Using theoretical tools from statistical mechanics, we show how and when does such representations emerge. In that case, RBM achieve a very good compromise between model expressivity and interpretability. We then present a new application for protein sequence modeling based on this principle.

In Part I, we introduce and illustrate through examples the key concepts required for this thesis: representations in machine learning, Boltzmann Machines (BM) and Restricted Boltzmann Machines (RBM). It is based on the following review article and currently unpublished material:

[33] S. Cocco, R. Monasson, L. Posani, S. Rosay, and J. Tubiana, "Statistical physics and representations in real and artificial neural networks," Physica A: Statistical Mechanics and its Applications, vol. 504, pp. 45–76, 2018.

Part II focuses on training algorithms for BM and RBM. We first review existing training methods, then introduce personal contributions developed over the course of my thesis. This part is based on the following article, currently under review:

[34] J. Tubiana and R. Monasson, "Efficient sampling and parametrization improve restricted boltzmann machines," 2018.

Part III is dedicated to the analysis of a model of RBM with random weights using statistical mechanics tools. After a review of network-based associative memory models and their statistical mechanics treatments, we present our model, present theoretical results and compare them to RBM trained on the MNIST handwritten digit data base. It is based on the following published article, as well as additional material to be published soon:

[35] J. Tubiana and R. Monasson, "Emergence of compositional representations in restricted boltzmann machines," Physical review letters, vol. 118, no. 13, p. 138301, 2017.

Finally, Part IV is dedicated to the application of RBM to protein sequence analysis. We start by reviewing major stakes of protein science, then present a short review of coevolution methods. It is based on the following two articles; the first is currently under review, and second will be submitted soon:

[36] J. Tubiana, S. Cocco, and R. Monasson, "Learning protein constitutive motifs from sequence data," arXiv preprint arXiv:1803.08718, 2018.

[37] J. Tubiana, S. Cocco, and R. Monasson, "Learning lattice proteins with restricted boltzmann machines : compositional regime and comparative analysis," 2018.

Part I

INTRODUCTION TO REPRESENTATIONS, BOLTZMANN MACHINES AND RESTRICTED BOLTZMANN MACHINES

2

REPRESENTATIONS

2.1 DEFINITION

We start with the definition of a data representation. Suppose we are given a set of P data samples $\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, ...\mathbf{x^{(P)}}$ of a N-dimensional random variable $\mathbf{X}$ having joint density $P(\mathbf{X})$. A data transformation is a deterministic transformation from the multidimensional vector space of data into another one:

$$F : \mathbf{x} \in \mathbb{R}^{\mathbf{N}} \rightarrow \mathbf{x'} = \mathbf{F}(\mathbf{x}) \in \mathbb{R}^{\mathbf{M}} , \tag{2.1}$$

where M can be larger or smaller than N. In general, F is assumed to be differentiable, but is not necessarily invertible. We say that the random vector $\mathbf{X'}$ is a representation of the original random vector $\mathbf{X}$. Changing the representation of a random variable can be often extremely helpful in data science because: i) it allows for better visualization and understanding of the process that generated the data; ii) the performance of machine-learning algorithm, such as classification or clustering methods heavily depends on the choice of representation used.

Although it is not obvious that a given representation is good, it is clear that many representations are useless: if $F(\mathbf{x}) = 0, \forall \mathbf{x}$, then $\mathbf{X'}$ is a trivial random variable, and does not carry any information about $\mathbf{X}$. More generally, it is clear that any transformation F that does not vary strongly across the support of $\mathbf{X}$ is of little use. On the opposite, $F = Id$ is not of much use either, since the properties of the data distribution have not changed. Typically, a good data representation $\mathbf{X'}$ must have helpful properties that $\mathbf{X}$ does not have, such as low dimensionality, independence between components or sparse values, while carrying information about the original random vector $\mathbf{X}$. Thus, the transformation F must depend on $P(\mathbf{X})$ and should be learnt. Once learnt, a data representation can often shed light on how the data was generated: one can find so-called 'features', i.e. frequent collective modes of variation in the data, find a partition into classes, discover outliers,...

One fundamental reason for learning new data representations is that the vector space $\mathbb{R}^N$ and its associated euclidian distance do not reflect well the

underlying structure of the data distribution. For instance, an image of an object and its copy translated by a few pixels are often very far away from one another in terms of euclidian distance - in fact, often as far away as two images of different objects. Similarly, in the context of protein sequences, it is well known that sequence similarity (the Hamming distance) is not always a good predictor of functional similarity. Moreover, the support of the data occupies only an infinitesimal fraction of the vector space, as data very often lie in or close to a subspace of dimension much lower than N. This is the so-called 'manifold hypothesis'. Indeed, consider for instance a data set constituted by pictures of a person's face, taken in many different positions; each picture is made of, say, 1000×1000 pixels. It is clear that this data set is a very small subset of all possible 1000×1000 colored pictures, which is defined by a $3 \cdot 10^6$–dimensional vector. The reason is that, for a given face, there are only ~ 50 varying degrees of freedom (the position of all muscles), a very small number compared to 10^6 [38]. Hence, all data points lie in a (non-linear) manifold, of very low dimension M compared to N. More generally, the variability in the data often comes from a small number of explanatory latent factors that affect all components, and we would like to recover them. In practice, the perfect representation algorithm that would turn an image into this kind of 'muscle positions' representation does not exist, because our problem is mathematically ill-defined. Indeed, given a set of latent factors (e.g. the 'true' set of muscle positions), any invertible transformation $\mathbf{z}' = G(\mathbf{z})$ also defines a set of latent factors that explains the variability in the data. A well-defined representation learning problem therefore requires making assumptions on the statistics and/or dimensionality of the latent factors, as well as on the transformation from factors to observations. We will present below some interesting representation learning algorithms based on these assumptions.

A good data representation can significantly improve the performance of subsequent machine learning tasks, by retaining only useful information about the data sample. For instance, in a so-called deep neural network, one learns a sequence of data transformations, e.g. to predict a label from an image. By using non-linearities and so-called pooling architectures, the learnt intermediate representations of the data can become invariant with respect noise, shifts, rotations,... hence learn quicker [17]. Deep neural networks led to remarkable breakthrough in many areas, such as visual and speech recognition, natural language processing [1, 39].

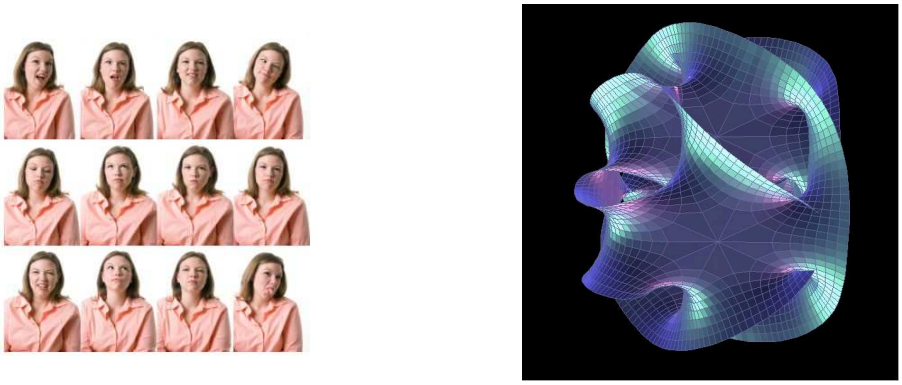

Figure 2.1: (a) Pictures of a person with various facial expressions. They lie in a very low dimensional manifold of the vector space of pictures with 1000×1000 pixels. (b) Example of complex 2D manifold embedded in a 3D vector space. Both examples are reproduced from Pr. Yann Lecun's lectures slides on Deep Learning (`https://www.slideshare.net/yandex/yann-le-cun`)

2.2 EXAMPLES

2.2.1 *Clustering and mixture models*

Arguably, the most simple representation of data is clustering. Clustering algorithms such as K-means or Dbscan identify subgroups within the data where the intra-cluster euclidian (or other) distance is low, and inter-cluster distance is high. Formally, it defines a deterministic mapping from the original data space $\mathbb{R}^N$ to a categorical variable $z \in [1, ..K]$. In the best cases, the subgroups identified by clustering are well separated and correspond to known categories, such as animal species in an image data base. In the worst cases, clusters found are unstable and do not relate to any known data structure. In particular, clustering depends on the choice of metric, and thus of the initial representation provided to the clustering algorithm.

Since allocation of a sample to a cluster can be ambiguous, probabilistic mapping $P(z = k|\mathbf{x})$ (so-called fuzzy clustering) can be derived instead, e.g. using a Gaussian Mixture Model. This defines a K-dimensional representation in which each dimension codes for a 'prototype' sample, and for most samples, a single component dominates over the others, see an example on MNIST, the handwritten digit data base in Fig. 2.2.

A. B.

Figure 2.2: A. Some samples of MNIST, the data set of 28×28 images of handwritten digits. B. Selection of prototypes inferred by a mixture model on MNIST. Each one is essentially an average of few digits from the data set.

2.2.2 *Dimensionality Reduction and PCA*

One important subclass of data transformations are dimensionality reduction transformations. One aims at compressing a random vector $\mathbf{X}$ of typically high dimension N, into a smaller random vector $\mathbf{X}'$ of dimension $M < N$, e.g. $M = 2$ or 3, while keeping as much information as possible about $\mathbf{X}$. Such compression is motivated by the manifold hypothesis described above. One example is Principal Component Analysis (PCA), in which dimensionality reduction is obtained through a simple linear transformation:

$$\mathbf{X}' = W\,\mathbf{X}\,. \tag{2.2}$$

where the weight W is a $M \times N$ rectangular matrix that must be trained on the data in order to retain as much information as possible from $\mathbf{X}$. This is done by minimizing a square error between the original data and the data reconstructed from the representation $\mathbf{X}'$. In practice, the PCA space is obtained by projecting the data onto the top M eigenvectors of the data covariance matrix $C_{ij} = \langle X_i X_j \rangle - \langle X_i \rangle \langle X_j \rangle$, where the average is computed over the data. Such transformation mainly serves two purposes. The first one is to provide a qualitative understanding of the data by visualizing it: one computes a 2 or 3 dimensional-representation of the data; then each data point is represented in a 2 or 3D space. For example, one can compute the 2D PCA representation of 28×28 images of digits from the MNIST handrwitten digits dataset, vectorized as 784-dimensional vectors, see Fig. 2.3; the scatter plot shows two distinct clusters, corresponding to two digit types (0s and 1s). A more interesting

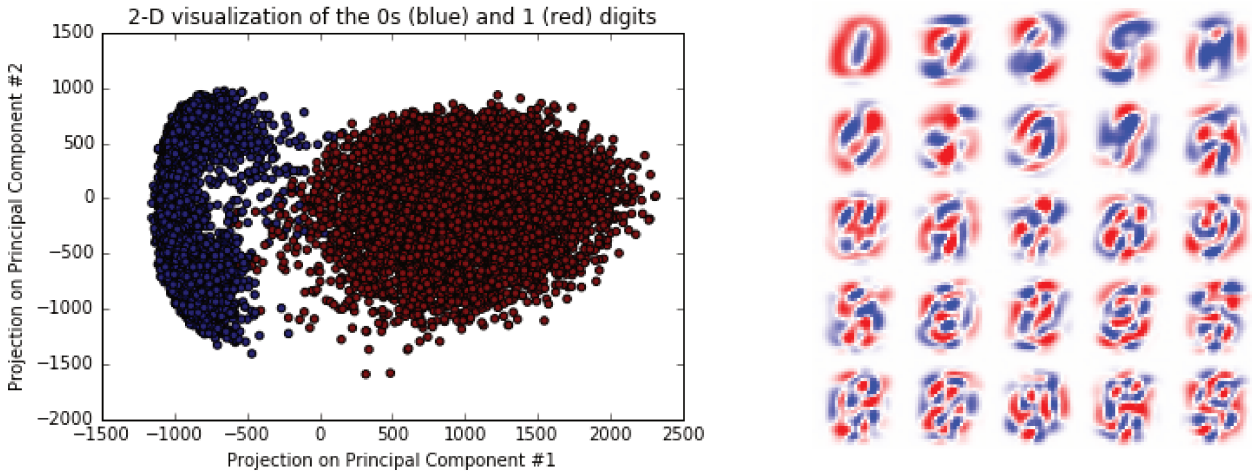

Figure 2.3: (a) A 2-dimensional PCA representation of the MNIST handwritten digits data set. Each point is a different image with x and y coordinates being the value of the first and second components of the representation. Here, only the digits 0's (blue) and 1's (red) are represented. (b) Visualization of the weight matrix W. Each image is a principal component vector $W_{i,.}$; blue (resp. red) pixels denote large positive (resp. negative) values. the PCA representation is obtained by computing the set of overlaps between an image and each principal component vector

illustration is the interpretation of molecular dynamics simulation of complex systems, made of many strongly interacting and heterogeneous microscopic components. Observing the dynamics of such systems, e.g. a protein described at the atomic level, amounts in practice to look at thousands of correlated time series. Principal component analysis offers low-dimensional projections of these time traces, and allows one to visualize collective motions underlying the evolution of the system, see [40] for a recent review on applications to biomolecules, including nucleic acids and proteins.

The second purpose of dimensionality reduction is to overcome the so-called curse of dimensionality. In very high dimensional spaces, most datasets sample only very sparsely the vector space $\mathbb{R}^N$. Consider for instance the following supervised learning problem. We are given a training data basis of $10,000$ 100×100 grayscale (normalized between 0 and 1) images of cats and dogs, with binary labels attached, and we want to train a parametric model to classify whether images are cats or dogs. At this point, it is useful to think that this classification task is essentially an interpolation problem: there exist a mathematical function $\theta : \mathbf{X} \to y \in \{0, 1\}$ that assigns 0 to cats and 1 to dogs.

We observe pairs of values $\left(\mathbf{X}^i, y^i = \theta(\mathbf{X}^i)\right)$, with $i = 1...10,000$, and want to interpolate the values of θ for new test images. This interpolation problem would be trivial if the input space was densely sampled, *e.g.* if for any point in $\mathbb{R}^N$ there would be a training data point at distance $\leq \epsilon$. In practice, it is impossible because the latter condition requires about ϵ^{-N} data points, which is out-of-reach when N is large.

One possible way-out is to first learn a new data representation of lower dimension, $\mathbf{x}' = \mathbf{F}(\mathbf{x})$, e.g. using PCA, and then train a classification model of the form: $y = \theta(\mathbf{x}')$. If the low dimensional representation keeps relevant information about the nature of the image, then learning can be performed. One popular application of PCA for supervised learning is the 'eigenface' face recognition algorithm. A PCA representation is trained on a data set of faces, before applying supervised learning [41]. The eigenface algorithm is considered among the first successful face recognition algorithms.

The main practical limitation of PCA is that it is generally difficult to identify the principal components with the latent factors mentioned above. As seen from Fig. 2.3(b), the weights are delocalized across all pixels, and cannot be related simply to the constitutents of digits. Weight delocalization is actually quite general: for any image data base featuring translational invariance, such as textures or natural images [42], the principal components are extended 2D Fourier modes [1]. In the next section, we briefly introduce discuss other feature extraction methods that aim at solving this issue.

2.2.3 *Extracting latent features from data*

The variability in real-world data, such as images, can often be decomposed into a set of largely independent modes of variation. For instance, two faces are different because some of their parts are different: nose, ears, lips... At a lower level of description, an image can contain or not an edge at a given location, or at some angle or scale, and two different images have different set of activated edges. Extracting these so-called 'features' is of particular interest for machine learning, in particular for classification, because the decision function $y = \theta(\mathbf{X})$ that must be learnt may be expressed more easily as a function of these 'features' $\mathbf{X}'$ than from the raw pixels $\mathbf{X}$. For instance, one could achieve better results by expressing $\theta(X')$ as a linear function of $\mathbf{X}'$, instead of a higher order polynomial

1 For instance, in the 1D case, a translational invariant data set yields a translational invariant covariant matrix of the form $C_{ij} = C(i-j)$. Assuming periodic boundary conditions, the eigenvectors are Fourier modes of the form $\lambda_j^k \propto e^{\frac{2i\pi k}{N}}$

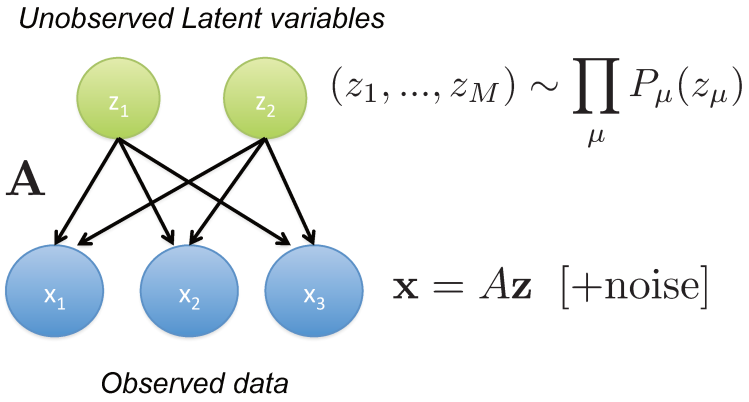

Figure 2.4: The linear mixing model. Observed random variable $\mathbf{X}$ is generated by first drawing latent variables z from a prescribed distribution, then linear transformation and addition of noise. Arrows indicate causal links

of $\mathbf{X}$. Moreover, the learnt representations have interesting statistical properties, such as low statistical dependence between modes, invariance with respect to irrelevant perturbations of the data such as corruption by noise. that can be used for denoising. Some notable algorithms for unsupervised feature extraction are Independent Component Analysis (ICA) [43], sparse dictionary learning [44] and sparse autoencoders [45].

Briefly, we assume in all cases the following directed graphical model, shown in Fig. 2.4, in which the latent factors $\mathbf{Z} \in \mathbb{R}^M$ are linearly mixed to produce the signal $\mathbf{X} \in \mathbb{R}^N$; the goal is to learn the mixing matrix. Since the mixing model is not unique (for any $\mathbf{Z}$, $\mathbf{W}$, $\mathbf{Z}' = \mathbf{U}\mathbf{Z}$ and $\mathbf{A}' = \mathbf{A}\mathbf{U}^T$ with $\mathbf{U}$ unitary induce the same distribution on $\mathbf{X}$), all methods rely on a probabilistic prior on the latent factor distribution. In the standard Infomax ICA framework [46], it is assumed that the z are non-gaussian and independent, there is no noise and $M \leq N$. The inference is carried out by finding $\mathbf{A}$ such that the z are as independent as possible from one another, using high order moments criteria. In the sparse dictionary framework, it is assumed that $M > N$ and the z are sparse, with a tunable sparsity prior λ. The inference is carried out by a double optimization process: for each sample $\mathbf{x}$, find the latent factor $\mathbf{z}$ that best compromises between reconstruction error and sparsity; then update $\mathbf{A}$ so as to reduce the reconstruction error and improve sparsity.

We display in Fig. 2.5 the features learnt by ICA on the MNIST digits data set. The features learnt correspond to individual handwritten strokes, i.e. parts of digits, unlike PCA where the principal component do not have a simple interpretation. Interestingly, the features found by sparse dictionary learning on natural images dataset qualitatively match very well the receptive fields of

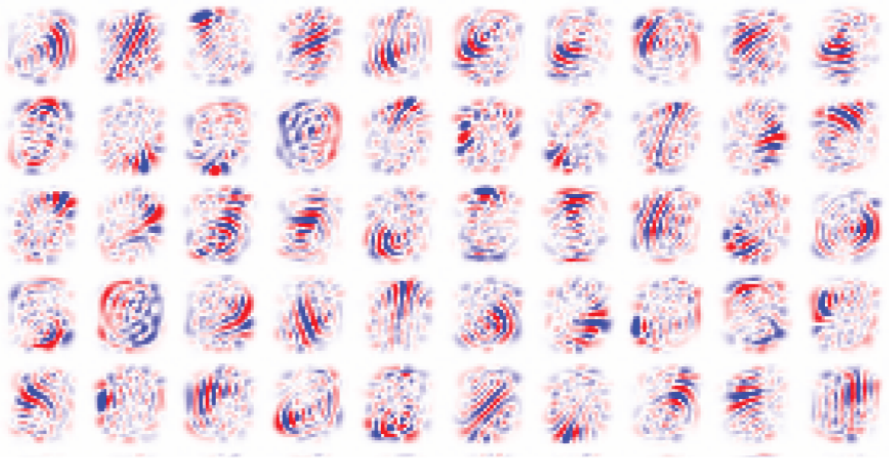

Figure 2.5: Features learnt by Independant Component Analysis on MNIST. Most features are localized around a region, and strongly activated by handwritten strokes, parts of digits. The Scikit-learn implementation of FastICA algorithm was used [50]

neurons in the visual cortex of mammalians, such as in monkey [47,48]. Feature extraction carried out in the brain bear strong analogies with machine-learning procedures [49].

ICA, sparse dictionaries and other linear blind source separations methods found numerous applications in the fields of signal processing and neuroscience. Sparse dictionaries inspired the development of new image restoration and compression algorithms [51,52], and led to the theory of compressed sensing (see below). In neuroscience, they were both successful as a theoretical framework for understanding the brain's visual processing [53] and for data analysis of functional recording [54,55]. Their main strength is their expressivity: owing to a large choice of latent feature activations, latent models can encode a diversity of samples with a limited number of non-zero entries. There are two main practical limitations. First, the choice of prior limits the set of latent factors that can be inferred. 'True' latent factors may not be independent; for instance, in images of faces, the latent factors 'is a man' and 'has a mustache' are correlated but distinct. Similarly, they may not be sparse, but rather binary, multimodal,... Second, linear mixing model is too simplistic, because the latent factors interact to generate the sample. For instance, translating an image or changing its angle of view is essentially equivalent to performing a permutation of edges detectors activations. Generally speaking, the latent factor - sample mapping is highly non-linear, and most modern approaches for latent factor inferring are based on deep architectures.

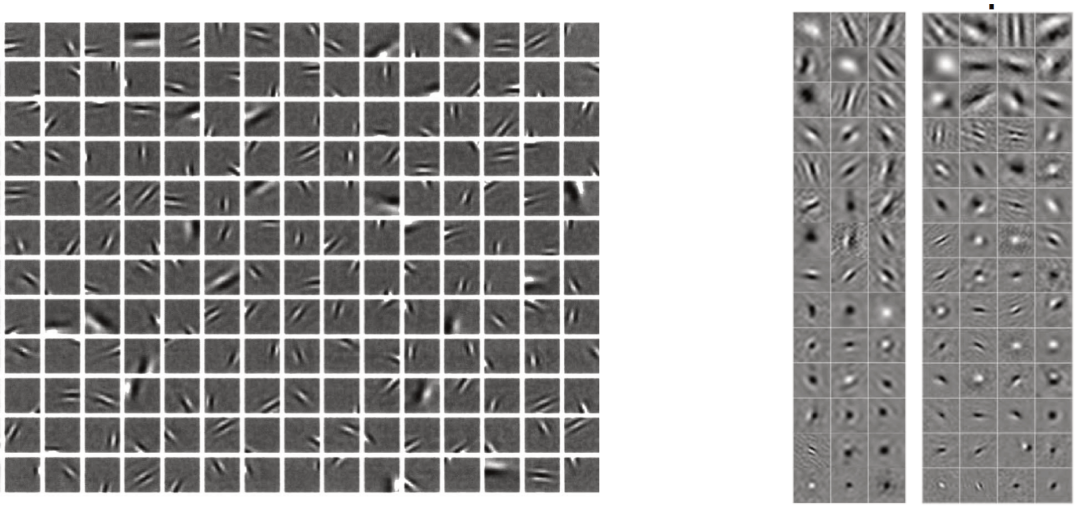

Figure 2.6: (a) Features learnt by Sparse Dictionary Learning applied to natural images (woods,...). Most features are orientation-specific edge detectors, reminiscent of Gabor filters. Picture reproduced from Olshausen and Fields [44] (b) Receptive fields of V1 simple cells in Macaque visual cortex, reproduced from [47]

2.2.4 *Random Projections and Compressed sensing*

One last interesting example of representation is the case of random projections, and its application to compressed sensing [56]. Here, we will assume that (i) there exists a known sparse dictionary such that $\mathbf{x} = \mathbf{A}\mathbf{s} \in \mathbb{R}^N$, where $\mathbf{s}$ has at most K non-zero entries for each $\mathbf{x}$ (ii) we compute a linear projection $\mathbf{x}' = W\mathbf{x}$, with a random matrix $\mathbf{W}$ in which each entry of the weights $w_{i\mu}$ is random, drawn from a gaussian distribution independently from the others. Provided that the number of measurements M is of the order of K (the exact boundary is given by the Donoho-Tanner phase transition [57]), it is possible to reconstruct accurately the original N dimensional signal, using e.g. LASSO optimization, see Fig. 2.7. Random projections are an extreme example of representation where all the information about the signal is present, but in a very intricate way; the individual components have no meaning on their own, and one must perform a complex non-linear transformation to recover the signal. In practice, many so-called incoherent bases, such as the Fourier representation in image processing share this property with random projections. Compressed sensing notably led to tremendous speed-up of MRI imaging [58].

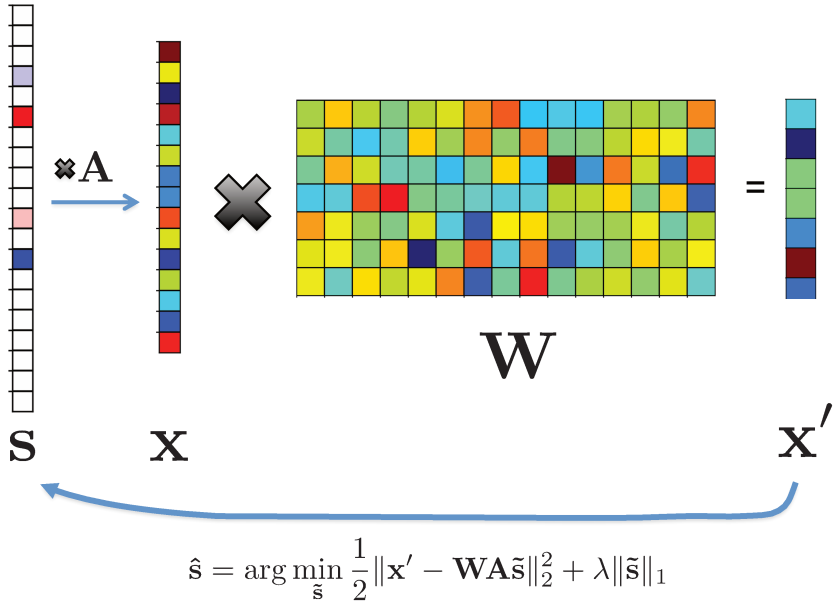

Figure 2.7: **Graphical summary of compressed sensing** A signal $\mathbf{x}$ is generated from a sparse latent feature with dictionary $\mathbf{A}$. A low-dimensional random projection is computed, from which reconstruction can be performed using LASSO reconstruction

2.3 SUMMARY

To summarize, we show in Fig. 2.8 the main types of representation:

- **Original representation**. Data typically forms a very sparse sampling of the high-dimensional space.
- **Prototypic representations**, such as clustering or mixture models. Each value or component encodes a specific portion of the data space, and for most samples, only one component is significantly active.
- **Intricate representations**, such as random projections. Representations are themselves high-dimensional vectors, whose component values are of similar amplitudes and all play similar roles across data space. Intricate representations can be very informative about the original data, but the individual components do not have simple interpretation.
- **Compositional representations**, such as sparse dictionaries. These representations are composed of elementary features that are activated by overlapping regions of the data space. Combining different activation patterns allows to describe a large diversity of data items all over the space.

Choosing one representation over the others ultimately depends on the kind of data and the subsequent intentions: it does not make sense to look for clusters when there are no natural classes within the data, or conversely to look for shared features in purely multimodal data. Moreover, not all representation learning algorithms can be allocated to one of the above mentioned types. We will show in particular that depending on the learning algorithm and the parameter values, RBM can behave in any of the last three representation types.

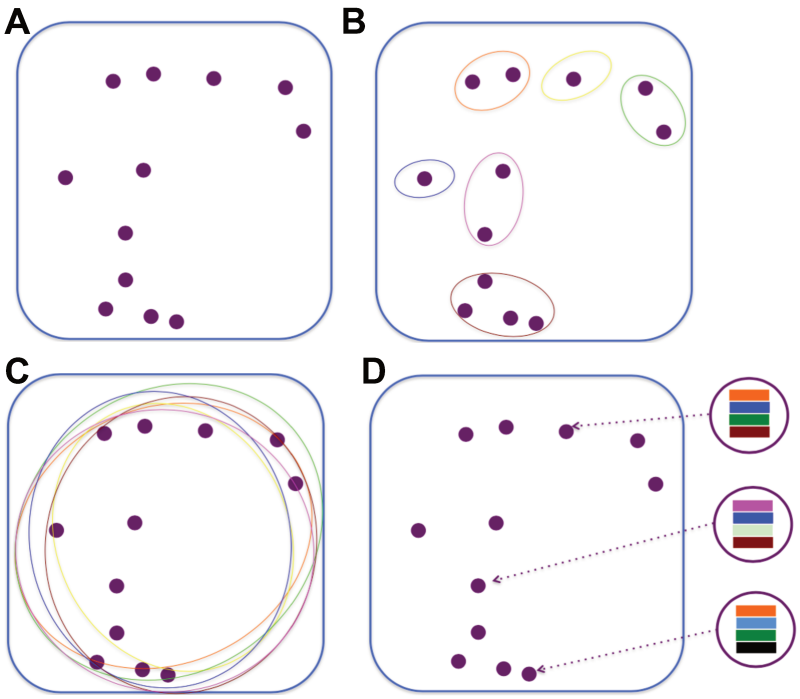

Figure 2.8: Nature of representations of data. A: Original data representation. B: Prototypic representation. C: Intricate representation. D: Compositional representation. Each dot represent a data sample, and each colored circle denote the region of the data space where a feature is significantly activated.

3

BOLTZMANN MACHINES AND RESTRICTED BOLTZMANN MACHINES

3.1 HISTORICAL BACKGROUND

Systems of interacting binary units were originally introduced as toy models of condensed matter systems in statistical physics. These coarse grained descriptions of interacting particles were designed as minimal models for studying collective phenomena in materials and phase transitions. Some famous examples include The Ising and Curie-Weiss model for studying ferromagnetism, paramagnetism and criticality in magnets [59,60], the Anderson model for studying conductor-insulator transition in materials [61,62], the Lebowitz-Penrose model of Liquid-Vapor phase transition [63], or the Sherrington-Kirkpatrick model of spin glasses [64].

These models were first brought to the to the domains of neuroscience and artificial intelligence in 80s, at the onset of the second wave of connectionism. In 1982, Hopfield showed that a system of coupled binary units mimicking a biological network of neurons connected by synapses could learn to store memories ('patterns'), and retrieve then under noisy conditions [65]. The main idea is to adjust the synapses such that each memory (a pattern of activation of the neurons) is an attractor of the dynamical system of interacting neurons; therefore, any dynamic starting around the attractor leads to the full retrieval of the pattern. The so-called Hopfield model of associative memory inspired a wide literature of attractor models in theoretical neuroscience [66]. In 1983, Ackley, Sejnowski and Hinton presented the Boltzmann Machine (BM), a system of coupled binary units whose biases and couplings could be trained by physics-like Monte Carlo simulations to learn implicit constraints from data [67]. BM were proven to be succesful for pattern completion tasks on toy examples, but the learning algorithm was prohibitively slow. In 1984, Geman and Geman showed a connection between Bayesian image denoising and bidimensional Ising-like lattice models of interacting spins: each spin plays the role of a pixel, and ferromagnetic couplings between neighbors arise from continuity priors in images. They then showed that the physics-inspired Gibbs sampling and simulated annealing were efficient for performing Maximum A Posteriori optimization

and denoising images. Finally, Smolensky presented the Harmonium - which is a special case of BM and is now known as Restricted Boltzmann Machine (RBM), in the context of the theory of language and symbolic computation [68].

Owe to their lack of computational efficiency, BM learning approaches were initially let down in favor of supervised learning algorithms based on backpropagation [69]. They reappeared in 2002 when Hinton proposed the Contrastive Divergence algorithm as a fast training algorithm for RBM [70], and RBM subsequently found interesting successes as representation learning algorithms [71] and for collaborative filtering (the Netflix problem) [72]. In particular, Hinton *et al.* showed that stacking RBM on the top of one another proved an efficient way of learning deep representations [73]; such deep belief networks could then be fine-tuned by backpropagation to reach state-of-the-art performance in classification tasks [74]. These results sparked a wave of interest in RBM [75], until the large amount of data, faster hardware and better regularizations rendered unsupervised pretraining of deep networks useless [1, 3, 76]. More recently, BM and RBM were also supplanted as generative models by new approaches, such as Variational Autoencoders [77] and Generative Adversarial Networks [14].

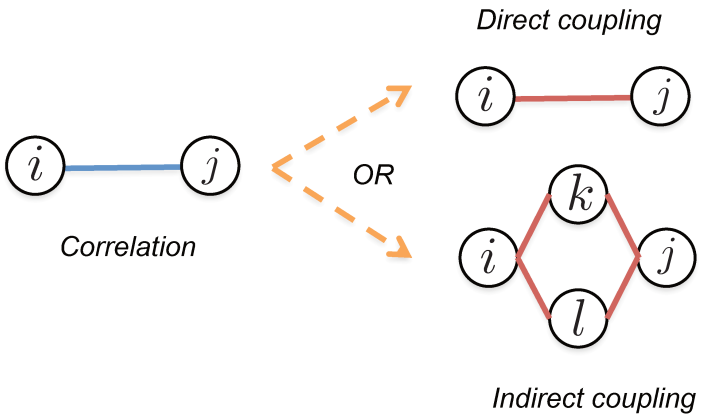

Figure 3.1: **The Inverse Ising problem** Correlations are either due to direct interaction or indirect interaction, as in the Ising model of statistical physics. The goal of the inverse Ising problem is to recover the interaction matrix from the data correlations

In parallel to these developments, Boltzmann Machines have witnessed a surge of interest in the statistical physics community over the last decade, under the names of inverse Ising problem or Maximum entropy modelling. In traditional statistical physics, macroscopic observables are derived from microscopic laws governing the system. In inverse problems the order is reversed: we are provided with observations of the system, and aim to go back to its microscopic laws. Inverse problems are particularly important for

studying complex systems in which the behaviour of individual units is well understood but not their collective behaviours. For instance, in neuroscience, the functional differentiation between various biological neural networks does not arise from neuronal types (there are only a few types across the whole brain), but rather by the way they interact: the set of axons, dendrites and synapses that mediates communication between neurons determines what computations are performed, how the network responds to external stimuli, and how it learns from experience. Each neural network has its own unique interaction graph, and it is essential to develop experimental or theoretical tools for elucidating network connectivity.

Since experimental measurement of all the synaptic couplings between all pairs of neurons is very challenging *in vivo*, recent approaches have focused on inferring them from observed neural activity only. The key idea is that a neuron receiving input from another excitatory or inhibitory neuron is respectively more or less likely to spike when the latter is spiking. Interactions between neurons therefore induce positive or negative spike correlations, and it may be possible to recover some information about the underlying network from the patterns of correlations. This can be formalized as an Ising inference problem: we look for a set of fields (the neural thresholds) and couplings (the synaptic interactions) that reproduces the mean and pairwise correlations from recordings. This problem is identical to learning a Boltzmann Machine with only visible units from the data. Numerous statistical physics methods were developed for solving the inverse Ising problem, such as message-passing algorithms [78], mean-field and TAP expansions [79,80], cluster expansions [81,82] see [32] for a review.

In the context of neuroscience, such approaches were shown successful at retrieving both structure of synaptic couplings and at predicting functional behavior (response to stimulus, replay, learning,...) in the retina [83,84], prefrontal cortex [85] and hippocampus [86–88], see [89,90] for reviews. Other examples of application of inverse Ising problem (and related models) include modelling of bird flocks [91], financial markets [92], and structure prediction in proteins or RNA (see Part iv), see [32] for a review. We now define BM and RBM and present result.

3.2 DEFINITION

A Boltzmann Machine (BM) and a Restricted Boltzmann Machine (RBM) are both undirected graphical models, i.e., probability distributions over a multidimensional space, defined via an interaction graph, see Fig. 3.2. BM are constituted of a single set of random variables $\mathbf{v}$, interacting via a coupling

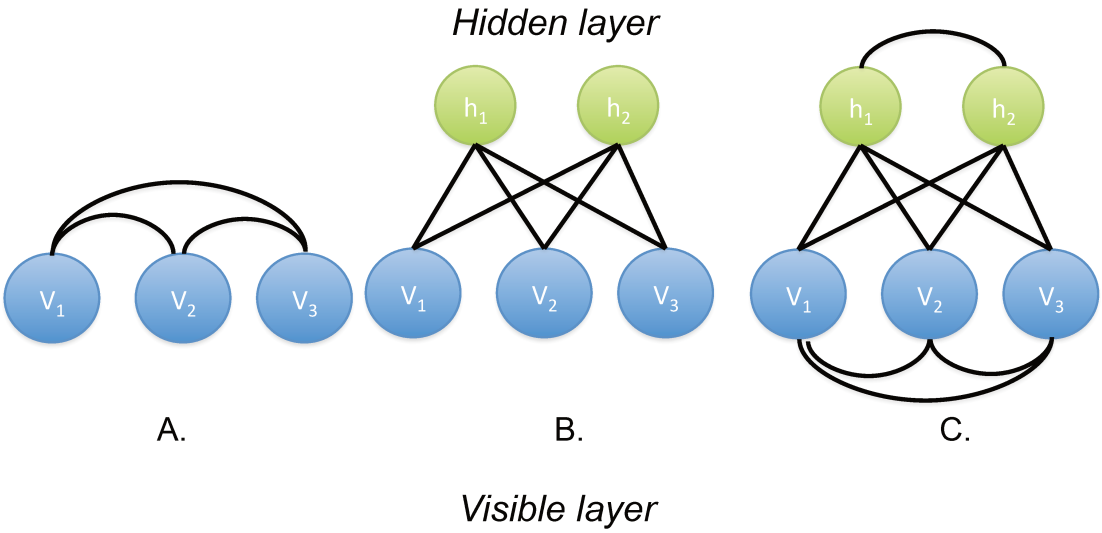

Figure 3.2: Architectures of BM (A), RBM (B) and gBM (C). All models are defined on a bidirectional graph; with a visible layer (v) representing the data and, for RBM and gBM a hidden layer (h) supposed to extract statistically meaningful features from the data.

matrix **J**. RBM are constituted by two sets of random variables, a visible layer (**v**) -the data layer- and a hidden layer (h), which are coupled together by a weight matrix **W**; there are no direct couplings between pairs of units in the same layer (hence, the name restricted). BM and RBM are both special cases of general Boltzmann Machines (gBM), originally formulated in [67], which have a visible layer, hidden layer and couplings between all pairs of variables. There are N visible units indexed by i, and for RBM and gBM, M hidden units indexed by μ.

In all generality, the variables v_i, h_μ can take binary (0/1 or -1/1), categorical (Potts state) or continuous ($\in \mathbb{R}$, $\mathbb{R}^+$, $[0,1]$,...) values; for simplicity of presentation we will assume here that the v_i are binary; generalization to other kind of variables is straightforward. For BM, the probability distribution of the visible unit configuration $\mathbf{v} = (v_1, v_2, ..., v_N)$ is given by the following Boltzmann distribution:

$$\begin{aligned} P(\mathbf{v}) &= \frac{1}{Z} e^{-E(\mathbf{v})} \\ E(\mathbf{v}) &= -\sum_{i=1}^{N} g_i v_i - \sum_{1 \le i < j \le N} J_{ij} v_i v_j \end{aligned} \tag{3.1}$$

Where E is the energy function and $Z = \sum_{\mathbf{v}\in[0,1]^N} e^{-E(\mathbf{v})}$ is the partition function such that P is normalized. The fields vector g_i and couplings matrix J_{ij} adjust respectively the mean and correlations of the units v_i. Similarly, for a RBM, the joint probability distribution of the visible and hidden unit configurations, $\mathbf{v} = (v_1, v_2, ..., v_N)$ and $\mathbf{h} = (h_1, h_2, ..., h_M)$ is:

$$\begin{aligned} P(\mathbf{v},\mathbf{h}) &= \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \\ E(\mathbf{v},\mathbf{h}) &= -\sum_i g_i v_i + \sum_{\mu=1}^{M} \mathcal{U}_\mu(h_\mu) - \sum_{i,\mu} w_{i,\mu} v_i h_\mu \end{aligned} \tag{3.2}$$

where as before, E is the energy function and $Z = \sum_{v,h} e^{-E(\mathbf{v},\mathbf{h})}$ is the partition function. $\mathcal{U}_\mu$ are unary potentials that control the marginal distributions of the variables h_μ, and the weight matrix $w_{i,\mu}$ couples the visible and hidden layers. The hidden potentials $\mathcal{U}_\mu$ can be chosen arbitrarily as long as sampling is feasible. Some useful examples are:

- The Bernoulli potential: $\mathcal{U}(x) = -gx \quad \text{with} \quad x \in \{0, 1\}$
- The Potts (multinomial) potential: $\mathcal{U}(x) = -g(x) \quad \text{with} \quad x \in \{1, .., K\}$
- The Quadratic or Gaussian potential: $\mathcal{U}(x) = \frac{1}{2}\gamma x^2 + \theta x, \quad \text{with } x \in \mathbb{R}$
- The ReLU potential: $\mathcal{U}(x) = \frac{1}{2}\gamma x^2 + \theta x, \quad \text{with } x \in \mathbb{R}^+$
- The double ReLU potential: $U(x) = \frac{1}{2}\gamma^+ x^{+2} + \frac{1}{2}\gamma^- x^{-2} + \theta^+ x^+ + \theta^- x^-, \; x \in \mathbb{R}$ where $x^+ = \max(x, 0)$, $x^- = \min(x, 0)$.

Bernoulli and Quadratic potentials are standard in the RBM literature; Potts potential is a straightforward generalization of RBM to categorical variables such as protein sites, with value 1 out of 20 amino-acids. The ReLU and double ReLU potentials were introduced during this thesis and will be justified below [1].

We stress that though the visible units are not directly connected, they are correlated thanks to common input from the hidden layer; RBM can therefore model correlated data. Indeed, consider the example of Fig. 3.3: a sample is collected from 4 binary variables that show strong Pearson correlations ~ 0.5 between all pairs. These samples could have been produced in two ways:

1 Nair and Hinton introduced ReLU for RBM in [93], but in an heuristic fashion: a conditional form $P(h_\mu|I_\mu) = \text{ReLU}\left(I_\mu + \mathcal{N}(0,1)\right)$ is prescribed, without any associated potential. The RBM were shown to be efficient for feature extraction, but cannot be used for scoring / generation purpose

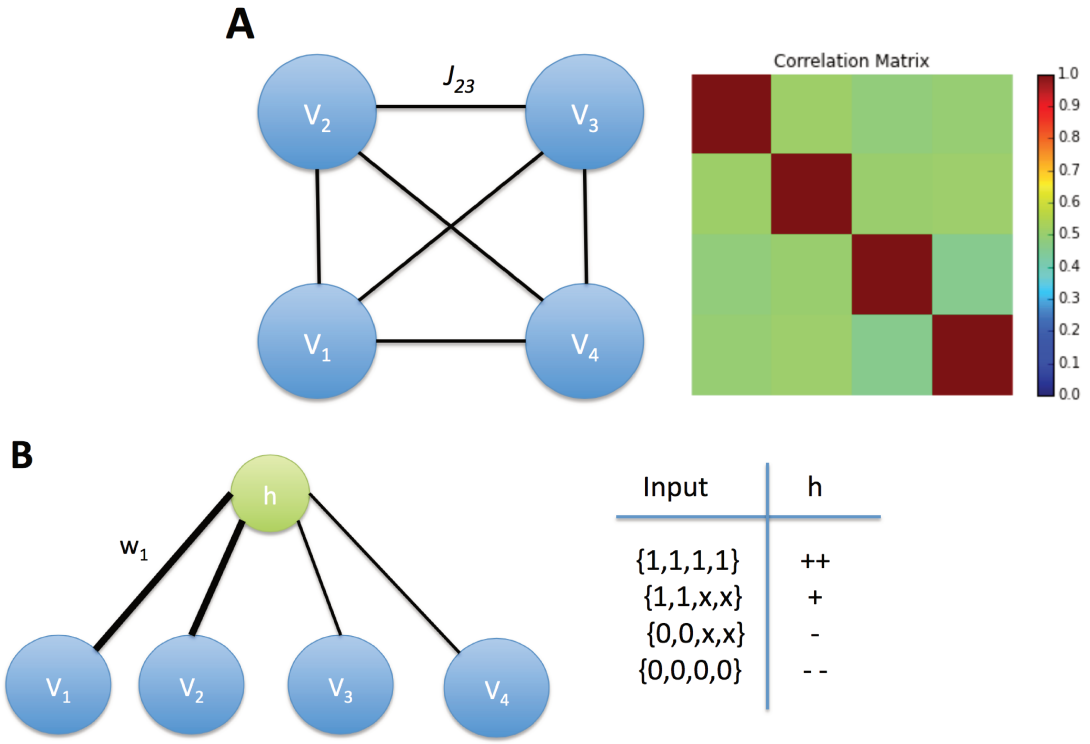

Figure 3.3: How to model correlations among a set of variables. **A.** Boltzmann Machine approach: The matrix of pairwise correlations between variables is computed from data, and a network of couplings is inferred to reproduce those correlations. **B.** Restricted Boltzmann Machine approach: observed correlations are due to one or more common input(s), whose values drive the configurations of the variables. A network of connection between the visible layer (support of data configurations) and a layer of hidden units (support of common inputs) is found to maximize the probability of the data items. The rightmost column indicates the magnitude h of the hidden unit as a function of the visible configuration.

- Either by a BM with excitatory couplings $J_{ij} > 0$ between all pairs of variables, such that a configuration with both $v_i = 1$ and $v_j = 1$ has lower energy (and higher probability) than a configuration with $v_i = 1, v_j = 0$ or $v_i = 0, v_j = 1$. Here, 'Ferromagnetic' interactions between each pair induce correlations.

- Either by a RBM with a single binary unit h_1, with positive coupling $w_{i1} > 0$ for all i. When $h_1 = 1$, each v_i has a high probability to be 1, $p(v_i = 1|h_1) = \frac{1}{1+e^{-(g+w)}}$ and conversely. In other words, the correlations between visible units arise come from a shared common input received by an unobserved unit.

Informally, the BM couplings represent causal links between units whereas the RBM hidden units represent collective modes of variation of the data. Formally, we can compute the probability distribution over the visible layer for RBM by marginalizing over the hidden units:

$$P(\mathbf{v}) = \int \prod_{\mu=1}^{M} dh_\mu P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\left(- \sum_{i=1}^{N} \mathcal{U}_i(v_i) + \sum_{\mu=1}^{M} \Gamma_\mu\big(I_\mu(\mathbf{v})\big) \right) \equiv \frac{1}{Z} e^{-E_{\text{eff}}(\mathbf{v})} \tag{3.3}$$

Where:

$$I_\mu(\mathbf{v}) = \sum_i w_{i\mu} v_i \tag{3.4}$$

is the input received by hidden unit μ, and:

$$\Gamma_\mu(I) = \log \left[\int dh \, e^{-U_\mu(h) + h\, I} \right] \tag{3.5}$$

is the cumulant generative function associated to the potential U_μ. For instance, for quadratic potential, $\Gamma_\mu(I) = \frac{1}{2\gamma}(I-\theta)^2 + \frac{1}{2}\log\frac{2\pi}{\gamma}$; if $\gamma_\mu = 1, \theta_\mu = 0$, the effective energy is, up to an additive constant:

$$E_{eff}(\boldsymbol{v}) = -\sum_i g_i \, v_i - \frac{1}{2} \sum_{i,j} \left(\sum_\mu w_{i\mu} w_{j\mu} \right) v_i v_j \tag{3.6}$$

In that case, we recognize a pairwise effective Hamiltonian with rank M pairwise interaction matrix, *i.e.* the Hopfield model with M patterns [65, 94]. In general, non-quadratic hidden-unit potentials have a non-quadratic cumulant generative function, and produce high-order interactions (obtained by Taylor of Γ). Interestingly, the number of high-order terms can be infinite but the number of parameters of the model is finite, scaling as MN; this is in stark contrast with high-order Boltzmann Machines, where each order adds $\mathcal{O}(N^k)$ parameters to the model. For both BM and RBM, training consists in fitting numerically the distribution $P(\mathbf{v})$ to the data by maximum likelihood, see section 3.4 and Part ii.

Thanks to its high-order interaction terms, RBM with Bernoulli hidden units are universal approximators, provided the number of hidden units is arbitrarily large [95]. In practice, the number of hidden units required can be relatively small or fairly large. For instance, the Curie-Weiss model, which is formally defined as a BM with i) ± 1 visible units ii) $g_i = 0 \ \forall i$ and iii) $J_{ij} = \frac{J}{N} \ \forall i, j$

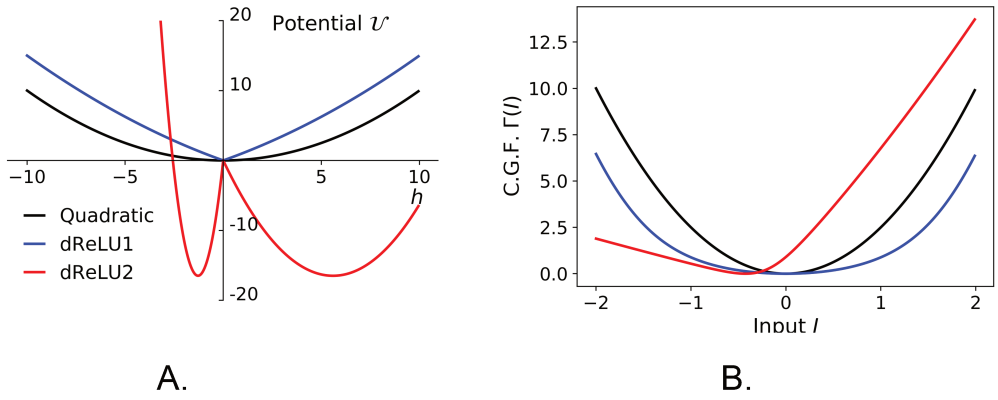

Figure 3.4: Hidden units potentials $\mathcal{U}_\mu$ and corresponding cumulant generative function Γ_μ, see Section 3.8 for analytical formula. Non-quadratic potential induce non-quadratic cumulant generative function and thus, high-order interaction terms.

with $J > 0$ can be expressed as a RBM with i) $M = 1$ hidden unit ii) a quadratic potential and iii) $w_{i1} = \sqrt{\frac{J}{N}}$; this is because $\mathbf{J}$ is positive definite of rank 1. On the other hand, the same model but with antiferromagnetic interactions $J < 0$ requires $N - 1$ hidden units, because $\mathbf{J}$ is negative definite. To view this, we can write the transformation $\mathbf{J} \to \mathbf{J} - J\mathbf{Id}$, which leaves the probability invariant but changes $\mathbf{J}$ into a positive definite matrix of rank $N - 1$. Qualitatively, antiferromagnetic interactions induce a narrower distribution of the average magnetization $m = \frac{1}{N}\sum_i v_i$ than with the independent model, and such constraint cannot be encoded simply by a shared input between visible units. More formally, this is because the cumulant generative function is always convex.

3.3 SAMPLING

In both BM and RBM, one cannot sample directly the distribution like in univariate distribution or multivariate gaussians. Instead, we must use Markov chain Monte Carlo sampling, using the Gibbs sampling to update configurations. Let $\mathbf{x}$ be the complete vector of units (either $\mathbf{v}$ or $(\mathbf{v}, \mathbf{h})$). Gibbs sampling consists in updating each unit x_l in a random order by drawing it from its conditional distribution $P(x_l|\mathbf{x}\ x_l)$. Gibbs sampling satisfies detailed balance, aperiodicity and in most cases irreducibility, such that given sufficient time, the Markov chain distribution converges toward the Boltzmann distribution. In RBM, the

connection with data representation algorithms is best seen when considering the sampling scheme. Since there are no connections within a layer, the hidden layer units are conditionally independent given the configuration of the visible layer, and conversely; hence the Gibbs sampling can be simplified as follows, schematized in Fig. 3.5:

- Compute hidden units inputs $I_\mu = \sum_i w_{i\mu} v_i$
- Sample each hidden unit independently $P(h_\mu|I_\mu) \propto \exp\left[-U_\mu(h_\mu) + h_\mu I_\mu\right]$
- Compute the visible layer inputs $I_i = \sum_\mu w_{i\mu} h_\mu$
- Sample each visible unit independently $P(v_i|I_i) \propto \exp\left[(g_i + I_i)v_i\right]$

The first two steps can be seen as a stochastic feature extraction from configuration $\mathbf{v}$, whereas the last two steps are a stochastic reconstruction of $\mathbf{v}$ from the features $\mathbf{h}$. One can define in particular a data representation as the most likely hidden layer configuration given a visible layer configuration, that is, through the set of

$$h^*_\mu(\mathbf{v}) = \arg\max_{h_\mu} P(h_\mu|\mathbf{v}) = H_\mu(I_\mu(\mathbf{v}))\ , \tag{3.7}$$

where $H_\mu = (\mathcal{U}'_\mu)^{-1}$ is the transfer function.

Another possiblity is to use the average hidden layer activity given the visible layer:

$$h^*_\mu(\mathbf{v}) = \langle h_\mu|\mathbf{v}\rangle \equiv \frac{\partial \Gamma_\mu}{\partial I}(I_\mu(\mathbf{v}))\ , \tag{3.8}$$

Where the last equality stems from the definition of the cumulant generative function (we have similarly $\mathrm{Var}(h_\mu|\mathbf{v}) \equiv \frac{\partial^2 \Gamma_\mu}{\partial I^2}(I_\mu \mathbf{v})$). We show in Fig. 3.6 examples of transfer functions and average activity. The nature of the hidden potential determines the shape of the transfer function and average activity. For quadratic potential, both are linear, whereas for Bernoulli potential, they are respectively a Heavyside and sigmoid function, see Section 3.8. For the Rectified Linear Unit (ReLU) potentials, the transfer function is exactly a ReLU function (hence the name) $H(I) = \mathrm{ReLU}(\frac{I-\theta}{\gamma})$, where $\mathrm{ReLU}(x) = \max(x, 0)$. ReLU is a popular non-linearity for neural networks, as they are easy to compute, can remove low signals by thresholding and do not saturate at large inputs, unlike sigmoids. For the double ReLU (dReLU) potential, the transfer function has two ReLU branches, see Fig. 3.6. Compared to Bernoulli hidden units, ReLU hidden units preserve information about the intensity of the input, and were shown to significantly outperform the former in the context of image recognition [93].

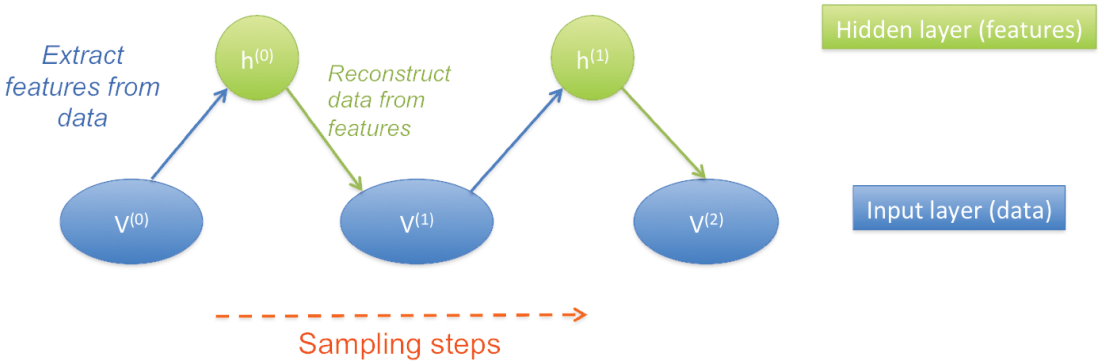

Figure 3.5: Back-and-forth sampling procedure in RBM. Hidden configurations $\mathbf{h}$ are sampled from visible configurations $\mathbf{v}$, and, in turn, define the distribution of visible configurations at the next sampling step.

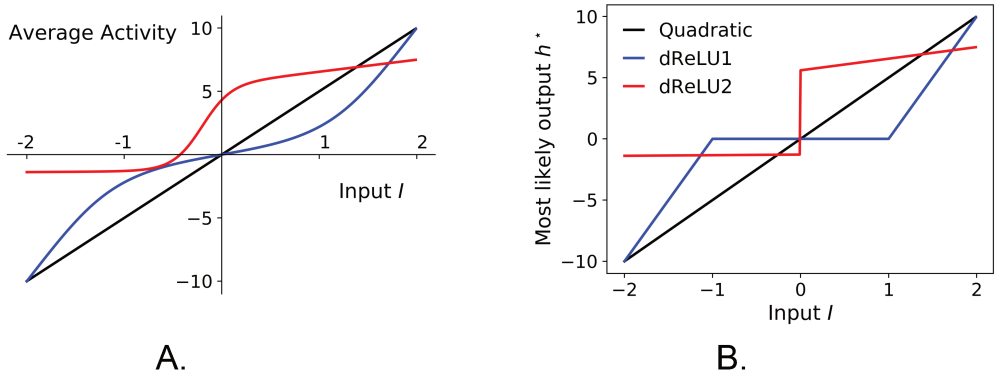

Figure 3.6: Average activity and transfer function for various hidden units potentials $\mathcal{U}_\mu$, see Section 3.8 for analytical formula. Non-quadratic potentials correspond to to non-linear transfer function and average activity

We have chosen a truncated gaussian distribution rather than followed the construction based on duplicating Bernoulli units proposed in [93] in order to obtain a well-defined probability distribution. As seen from Fig. 3.6, the dReLU potential, which we introduced in [36] is the most general form, and can effectively interpolate between quadratic ($\theta_+ = \theta_-$, $\gamma_+ = \gamma_-$), ReLU ($\gamma_- \to \infty$), and Bernoulli ($\gamma_\pm = \mp\theta_\pm \to +\infty$) potentials. In practice, the parameters are adjusted during training and this flexibility is particularly useful for modeling both super-gaussian and sub-gaussian projections, see Section 3.7.

3.4 LEARNING

Training is achieved by maximizing the likelihood of the data $\mathcal{L} = \langle \log P(\mathbf{v}) \rangle_d$. For any general parametric Boltzmann distribution $P_\theta(\mathbf{v}) = \frac{1}{Z} e^{-E(\mathbf{v},\grave{})}$, the gradient with respect to $\grave{}$ is given by:

$$\nabla_{\grave{}} \mathcal{L} = - \langle \nabla_{\grave{}} E(\mathbf{v},\grave{}) \rangle_d + \langle \nabla_{\grave{}} E(\mathbf{v},\grave{}) \rangle_m \tag{3.9}$$

Where $\langle \rangle_m$ is the expectation over the current model distribution $P_\theta(\mathbf{v})$. For a Boltzmann Machine, this gives:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial g_i} &= \langle v_i \rangle_d - \langle v_i \rangle_m \\ \frac{\partial \mathcal{L}}{\partial J_{ij}} &= \langle v_i v_j \rangle_d - \langle v_i v_j \rangle_m \end{aligned} \tag{3.10}$$

And for a Restricted Boltzmann Machine, with hidden unit potential U_μ and associated potential parameters ξ_μ (e.g. fields, threshold curvature,...):

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial g_i} &= \langle v_i \rangle_d - \langle v_i \rangle_m \\ \frac{\partial \mathcal{L}}{\partial \xi_\mu} &= \left\langle \frac{\partial \Gamma_\mu(I_\mu(\mathbf{v}))}{\partial \xi_\mu} \right\rangle_d - \left\langle \frac{\partial \Gamma_\mu(I_\mu(\mathbf{v}))}{\partial \xi_\mu} \right\rangle_m \\ \frac{\partial \mathcal{L}}{\partial w_{i\mu}} &= \left\langle v_i \left\langle h_\mu | \mathbf{v} \right\rangle \right\rangle_d - \left\langle v_i \left\langle h_\mu | \mathbf{v} \right\rangle \right\rangle_m \end{aligned} \tag{3.11}$$

Where we used the identity $\langle h_\mu | \mathbf{v} \rangle = \frac{\partial \Gamma_\mu(I_\mu(\mathbf{v}))}{\partial I}$. In all cases, the gradient is the difference between a moment from the data and its corresponding moment of the model distribution; at the maximum of likelihood, moment matching conditions are satisfied. The major difficulty lies in evaluating the second term $\langle f(\mathbf{v}) \rangle_m = \frac{\sum_{\mathbf{v}} f(\mathbf{v}) e^{-E(\mathbf{v})}}{\sum_{\mathbf{v}} e^{-E(\mathbf{v})}}$, because it involves a weighted summation over 2^N configurations, which is impossible in practice. Beyond the computational difficulty, estimating the moments from interactions is fundamentally difficult, because as is known in statistical physics, very small changes in interaction parameters can induce phase transitions *i.e.* dramatic changes in the moments.

We discuss in Part ii current moment approximation methods, and present a new sampling algorithm, as well as new dynamic reparameterization techniques for addressing these issues.

The main differences between training BM and RBM are that:

- in BM, the data moments can be evaluated only once, whereas they must be recomputed as **W** evolves in RBM. Regular gradient descent is therefore best suited for BM, whereas stochastic gradient descent is faster for RBM.
- Sampling is slightly easier for RBM due to the conditional independence property, which allows parallel updates instead of sequential ones.
- RBM can be less computationally demanding, as one can choose $M \ll N$ (iv) the likelihood is a convex function for BM but not for RBM.

3.5 LIKELIHOOD ESTIMATION

Since the partition function Z is intractable in both BM and RBM, the log-likelihood $\mathcal{L} = \log P(\mathbf{v})$ cannot be computed directly. Throughout this manuscript, we have used the Annealed Importance Sampling (AIS) algorithm for estimating the partition function and therefore the likelihood [96, 97]. Briefly, the idea is to estimate partition function ratios. Let $P_1(\mathbf{x}) = \frac{P_1^*(\mathbf{x})}{Z_1}$, $P_0 = \frac{P_0^*(\mathbf{x})}{Z_0}$ two probability distribution with partition functions Z_1,Z_0. Then:

$$\left\langle \frac{P_1^*(\mathbf{x})}{P_0^*(\mathbf{x})} \right\rangle_{\mathbf{x}\sim P_0} = \sum_{\mathbf{x}} \frac{P_1^*(\mathbf{x})}{P_0^*(\mathbf{x})} \frac{P_0^*(\mathbf{x})}{Z_0} = \frac{1}{Z_0} \sum_{\mathbf{x}} P_1^*(\mathbf{x}) = \frac{Z_1}{Z_0} \tag{3.12}$$

Therefore, provided that Z_0 is known (e.g. if P_0 is an independent model with no couplings), one can in principle estimate Z_1 by Monte Carlo. The difficulty lies in the variance of the estimator: if P_1, P_0 are very different from one another, some configurations can be very likely for P_1 and almost impossible for P_0; these configurations appear almost never in the Monte Carlo estimate of $\langle . \rangle$, but the probability ratio can be exponentially large. In Annealed Importance Sampling, we address this problem by constructing a continuous path of interpolating distribution $P_\beta = P_1^\beta P_0^{1-\beta}$, and estimate Z_1 as a product of ratios of partition function:

$$Z_1 = \frac{Z_1}{Z_{\beta_{l_{max}}}} \frac{Z_{\beta_{l_{max}-1}}}{Z_{\beta_{l_{max}-2}}} ... \frac{Z_{\beta_1}}{Z_0} \times Z_0 \tag{3.13}$$

In practice, we choose P_0 as the closest (in terms of KL divergence) independent model to the data distribution P_d, and a linear set of interpolating temperatures of the form $\beta_l = \frac{l}{l_{\max}}$. To evaluate the successive expectations, we use a fixed number M of samples initially drawn from P_0, and gradually anneal from P_0 to P_1 by successive applications of Gibbs sampling at P_β. Moreover, all computations are done in logarithmic scales for numerical stability purposes: we estimate $\log \frac{Z_1}{Z_0} \approx \left\langle \log \frac{P_1^*(\mathbf{x})}{P_0^*(\mathbf{x})} \right\rangle_{\mathbf{x}\sim P_0}$, which is justified if P_1 and P_0 are close. We refer interested readers to [97] for implementation details.

3.6 RESULTS ON MNIST

We show in Fig. 3.7 and 3.8 selected results of training of respectively BM and RBM with various potentials on the MNIST digits data set. For BM, Fig. 3.7A shows a selection of pixel-pixel couplings; each image corresponds to a coupling $J_{i.}$ at fixed pixel i (shown in red). The couplings are mostly non-zero in the neighborhood of the pixel, featuring short range excitation and intermediate range inhibition. Crucially, the distribution of coupling values is much sparser than the distribution of correlations (panel B). This is because a large fraction of the correlations C_{ij} are indirect and can be explained by other couplings J_{ik}, J_{jk}, see Fig. 3.1: in interacting systems such as the Ising model, long-range correlations can arise from local couplings.

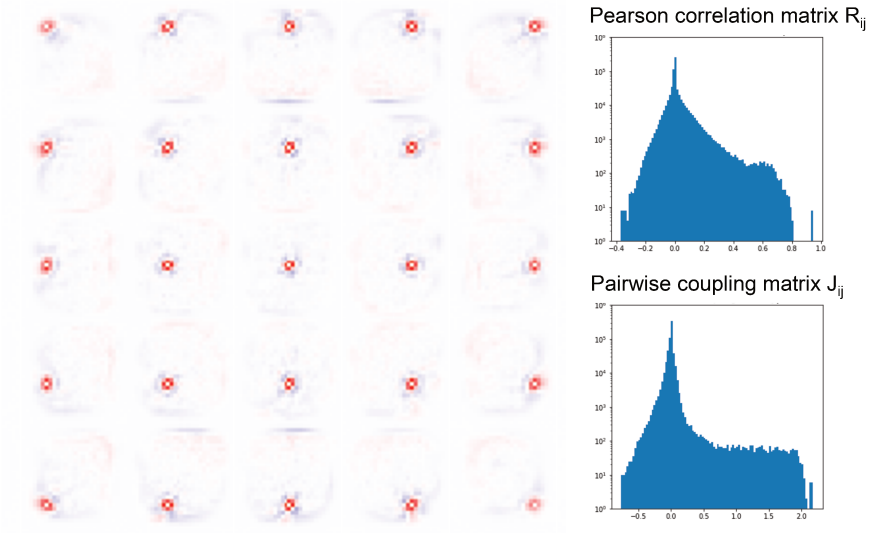

Figure 3.7: **Modeling MNIST with Boltzmann Machines** A. Visualization of the interaction matrix J_{ij} inferred. Each image shows the couplings $J_{i.}$ for a fixed i, identified by the white pixel. Direct couplings are most important between close neighbors B. Distribution of the Pearson correlation coefficients. Distribution of the inferred coupling values

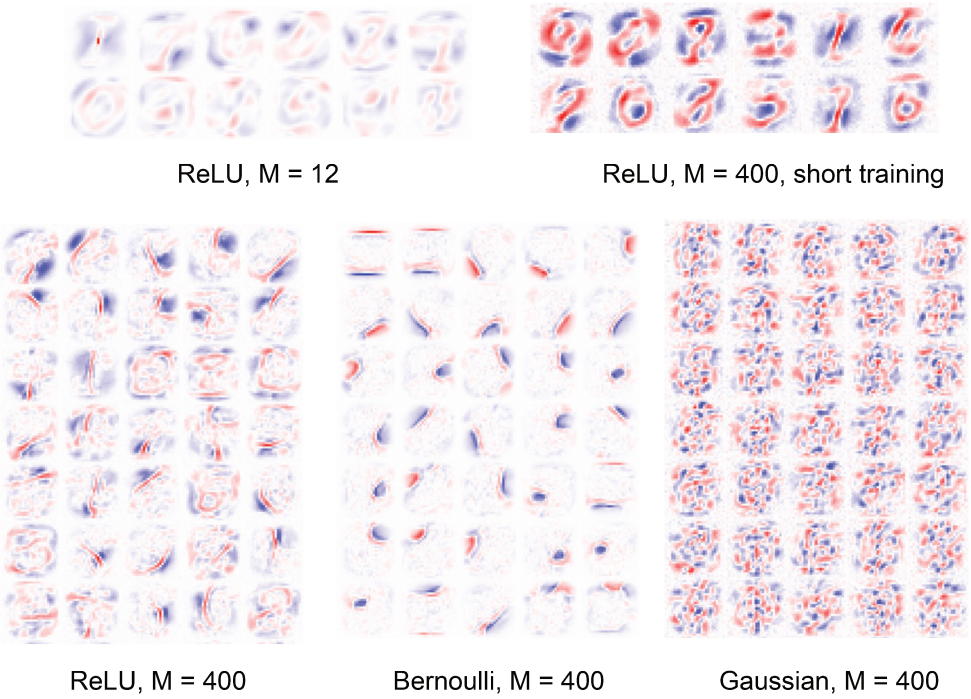

Figure 3.8: **Modeling MNIST with Restricted Boltzmann Machines** Selection of features inferred by RBM, with $M = 12$ or $M = 400$, and with ReLU, Bernoulli or Gaussian hidden unit potentials. We find prototypes, features or extended modes.

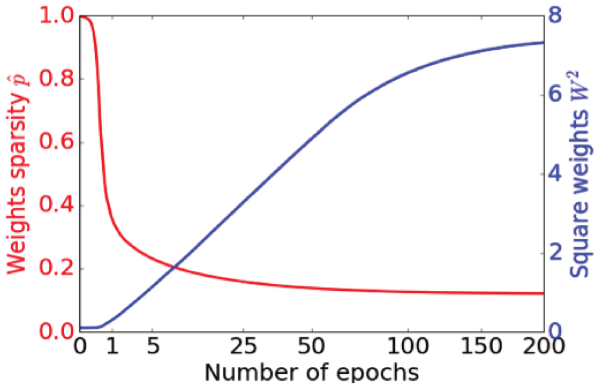

Figure 3.9: **Weight evolution throughout training** For ReLU RBM with $M = 400$, as training converges, the weights become sparser and larger.

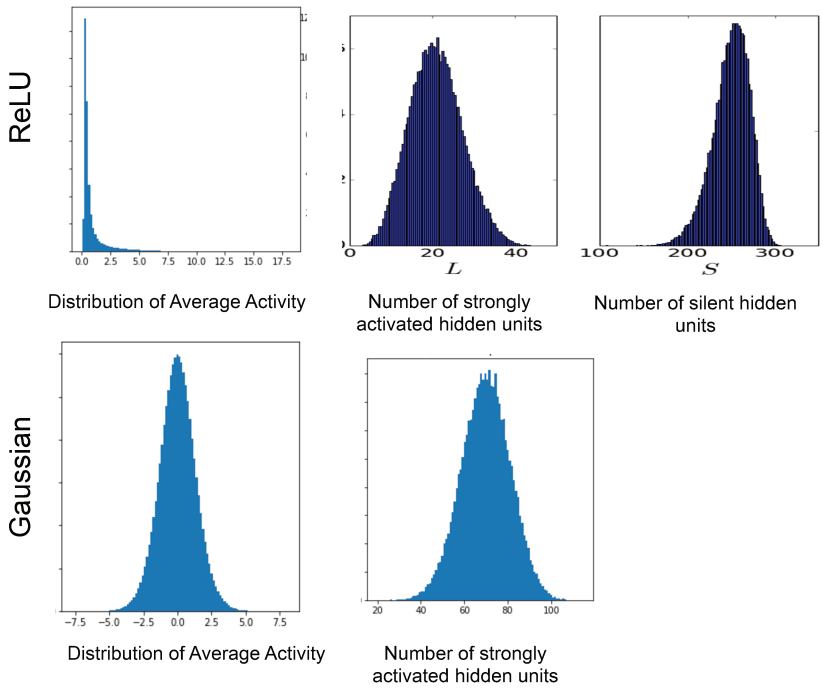

Figure 3.10: **Properties of learnt representations** Distribution of the hidden layer average activities $\langle h_\mu | \mathbf{v} \rangle$ (left), of the number of significantly activated hidden units (middle) and silent units (right), for ReLU (top) and Gaussian (bottom) potentials with $M = 400$ and regular training.

The phenomenology is richer for RBM. We show in Fig. 3.8 a selection of features inferred using Bernoulli, Gaussian and ReLU hidden units, with $M = 400$ or $M = 12$, and short or regular training. The shape of the features and the nature of the representation depends on the training time, the number of hidden units and the nature of the potential. For ReLU potential (and Bernoulli,

not shown) and i) low M or ii) large M and short training time, each weight is extended, with a shape similar to a single digit, as was observed with the mixture model, see Fig.2.2. For Bernoulli and ReLU potentials with large M, as training converges, the weights get larger and sparser, reaching a fraction of nonzero weights $p \sim 0.1$, see Fig. 3.9. Each weight is localized, and encodes a stroke - as was found by ICA, see Fig. 2.5. The hidden layer representation is also similar to sparse dictionaries, showing a compositional behavior: each data image strongly activates a small number of hidden units $L \sim 20$ (see Part iii for mathematical definition of p and L), whereas most hidden units are silent (I below ReLU threshold) or weakly activated, see Fig. 3.10. On the other hand, for Gaussian potentials, the features are delocalized, and seemingly unrelated to the image data base. Histograms of hidden average hidden unit activity h_μ are Gaussian, which is typical of intricate representations such as random projection (see Section 2.2.4). This can be explained by the fact that the marginal probability distribution $P(\mathbf{v})$ is invariant under a rotation of the weights $\mathbf{W} \to \mathbf{WO}$, where O is a rotation matrix with $O^T O = Id$, see Eqn. (3.6). There is therefore a large degeneracy of equally performing models, and most have intricate representations.

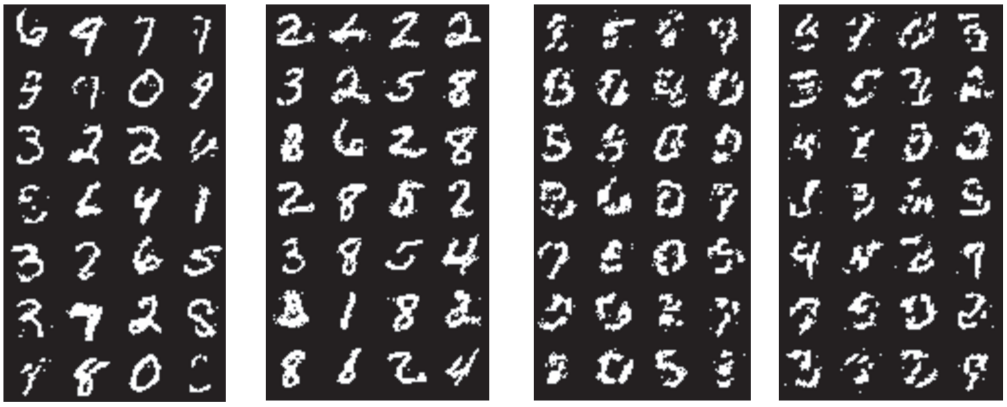

Figure 3.11: **Artificially generated digits** Samples from the various models and likelihood per pixel. See Fig. 2.2 for a comparison with digits from the original data base.

The match between the data distribution and the inferred probability distribution can be assessed by the quality of generated samples: for good matches, samples from the probability distribution should look like digits. Fig. 3.11 shows samples from all 4 distributions. We find decent looking digits for Bernoulli and ReLU RBM, but not BM and Gaussian RBM. Overall, this suggests that high-order moments are crucial both for the nature of representation

and for generative purpose. Log-likelihood estimates obtained by Annealed Importance Sampling (see section 3.5) confirm the visual impression, showing higher log-likelihood for high-order models than pairwise models.

Interestingly, the samples of ReLU RBM are also very diverse: the learnt probability distribution is very complex, with many local maxima of probability (much larger than the values of N or M) as seen in Fig. 3.12. For each sample from the training and testing set, we perform a gradient ascent on $P(\mathbf{v})$ to find its closest attractor (i.e. local maximum of probability); we then count the number of distinct attractor at various phase of the training. At the beginning, the model is monomodal, with a single attractor. As training progresses, the number of attractors grows. After training, each data sample - both from training and test set - is within few pixels of a distinct local maximum of probability. Beyond the tested samples, the total number of attractors of the model is likely very large. Intuitively, this is because different combinations of activated features produce different visible layer configurations, such that combinatorics can create large sample diversity.

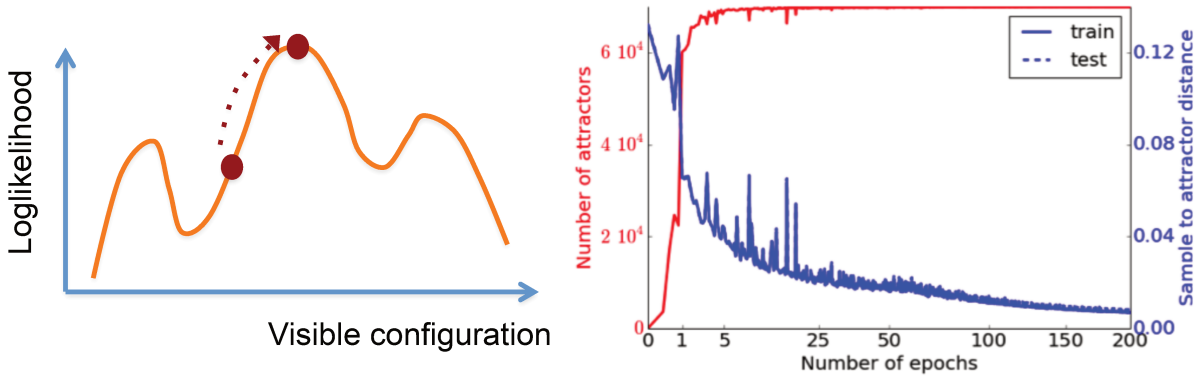

Figure 3.12: **Probability landscape learnt on MNIST** Counts of the number of distinct attractors (local maxima of $P(\mathbf{v})$ around the data samples. After training, each sample is very close (7-8 pixel) to a single attractor, suggesting a very rough landscape.

Overall, depending on the choices of potentials, training time and number of hidden units, we found a diversity of learnt representations, model performances and probability landscapes. In particular, RBM switch between prototype, compositional and intricate representations. The motivation of Part iii is to better understand why these different representations occur.

3.7 WHAT DO HIDDEN UNITS LEARN?

In BM, it is fairly clear that the couplings are adjusted so as to match the data and model correlations. On the other hand, the moments $v_i \langle h_\mu | \mathbf{v} \rangle$ adjusted by RBM depend on the weight matrix $\mathbf{W}$ and on the non-linearity chosen, and they are dynamically evolving throughout training. What are the hidden units trying to model ? Here, we present a new short computation illustrating the process. In the following we assume that we sequentially add each hidden unit and update only its parameters $\mathbf{w}_\mu, \xi_\mu$ (and not the previous ones $\mathbf{w}_{\mu'}, \xi_{\mu'}$ $\mu' < \mu$) instead of performing a gradient descent over the entire set of parameters. Moreover, we will assume for now that the cumulant generative function $\Gamma_\mu(I)$ is arbitrary rather than parametric, *i.e.* such that its value for each I can be adjusted. To this end, let P_μ be the marginal probability distribution over the visible layer where only the first μ hidden units are included:

$$\log P_\mu(\mathbf{v}) = g^T v + \sum_{\mu'=1}^{\mu} \Gamma_{\mu'}(I_{\mu'}) - \log Z_\mu \tag{3.14}$$

Where $Z_\mu = \sum_{\mathbf{v}} e^{g^T v + \sum_{\mu'=1}^{\mu} \Gamma_{\mu'}(I_{\mu'})}$ is the associated partition function. The following recursion relation holds:

$$\begin{aligned} P_\mu &= P_{\mu-1} e^{\Gamma_\mu(I_\mu)} \frac{Z_{\mu-1}}{Z_\mu} \\ &= P_{\mu-1} e^{\Gamma_\mu(I_\mu) - \log \left\langle e^{\Gamma_\mu(I_\mu)} \right\rangle_{P_{\mu-1}}} \end{aligned} \tag{3.15}$$

For $\mu = 0$, P_0 is an independent model with fields g. Initially, we set the fields $g_i(a) = \log f_i(a)$, *i.e.* the fields of the independent model closest to the data. Then, the RBM is recursively built: given $\{\mathbf{w}_{\mu'}, \Gamma_{\mu'}, \mu' \in [1, \mu-1]\}$, we derive w_μ, Γ_μ by maximum likelihood estimation. From Eqn. 3.15, the likelihood writes:

$$\mathcal{L}_\mu = \langle \log P_\mu(\mathbf{v}) \rangle_d = \langle \log P_{\mu-1}(\mathbf{v}) \rangle_d + \langle \Gamma_\mu(I_\mu) \rangle_d - \log \left\langle e^{\Gamma_\mu(I_\mu(\mathbf{v}))} \right\rangle_{P_{\mu-1}} \tag{3.16}$$

Where the first term does not depend on Γ_μ, $\mathbf{w}_\mu$. Deriving first with respect to $\Gamma_\mu(I)$ yields:

$$\begin{aligned}\frac{\delta \mathcal{L}_\mu}{\delta \Gamma_\mu(I)} &= \left\langle \delta(I - I_\mu) \right\rangle_d - \frac{\left\langle \delta(I - I_\mu) e^{\Gamma_\mu(I_\mu(\mathbf{v}))} \right\rangle_{P_{\mu-1}}}{\left\langle e^{\Gamma_\mu(I_\mu(\mathbf{v}))} \right\rangle_{P_{\mu-1}}} \\ &= P_d(I_\mu = I) - e^{\Gamma_\mu(I) - \log \left\langle e^{\Gamma_\mu(I_\mu(\mathbf{v}))} \right\rangle_{P_{\mu-1}}} P_{\mu-1}(I_\mu = I)\end{aligned} \tag{3.17}$$

Where $P_d(I_\mu)$ (resp. $P_{\mu-1}(I_\mu)$) denote the induced probability density of the input I_μ under $P_d(\mathbf{v})$ (resp. $P_{\mu-1}(\mathbf{v})$). Solving for the critical point gives Γ_μ up to an additive constant:

$$\Gamma_\mu(I) = \log \left[\frac{P_d(I_\mu = I)}{P_{\mu-1}(I_\mu = I)} \right] + K \tag{3.18}$$

The choice $K = 0$ is convenient, as it gives $\log \left\langle e^{\Gamma_\mu(I_\mu(\mathbf{v}))} \right\rangle_{P_{\mu-1}} = 0$. P_μ is given by:

$$P_\mu(\mathbf{v}) = P_{\mu-1}(\mathbf{v}) \frac{P_d(I_\mu(\mathbf{v}))}{P_{\mu-1}(I_\mu(\mathbf{v}))} \tag{3.19}$$

Note also that Γ_μ is such that $P_\mu(I_\mu) = P_d(I_\mu)$. Intuitively, for a fixed $\mathbf{w}$-, Γ_μ is adjusted such that the histograms of I_μ under P_μ and P_d match. After optimizing of Γ_μ, the likelihood 3.16 rewrites:

$$\begin{aligned}\mathcal{L}_\mu &= \left\langle \log P_\mu(\mathbf{v}) \right\rangle_d = \left\langle \log P_{\mu-1}(\mathbf{v}) \right\rangle_d + \left\langle \log \frac{P_d(I_\mu(\mathbf{v})}{P_{\mu-1}(I_\mu(\mathbf{v})} \right\rangle_d \\ &= \mathcal{L}_{\mu-1} + D_{KL}\left(P_d(I_\mu) || P_{\mu-1}(I_\mu) \right)\end{aligned} \tag{3.20}$$

Where D_{KL} denotes the Kullback-Leibler (KL) divergence. Hence, maximizing over $\mathbf{w}_\mu$ amounts to finding the linear projection that maximizes the KL divergence between the data distribution and the previous model distribution. In other words, hidden unit μ first finds the most discriminating feature between the target distribution and the current distribution $P_{\mu-1}$ (in a very similar spirit

to the discriminator in GANs [14]), then it is incorporated to the model and its potential is adjusted in order to exactly erase this difference. Since $D_{KL} \geq 0$, the process always increases the likelihood and is therefore guaranteed to converge to a local maximum. In practice, the cumulant generative function is not arbitrary but parametric; this biases the search of projections toward particular statistics: Bernoulli potentials favor projections with bimodal distributions, and so on. dReLU potentials, which can express all symmetric and asymmetric distributions, and gaussian, sub-gaussian or super-gaussian distributions (*i.e.* bimodal or sparse) are the least biased potentials.

This iterative scheme is very similar to the process of finding the top K principal components of the data: one computes the data covariance matrix, then looks for the component with highest variance (the top eigenvector), substracts it to the covariance matrix, and repeats the cycle. The main differences are that (i) RBM aim at explaining data probability, whereas PCA solely explains variance and (ii) the iterative procedure described above gives the best possible result for PCA, but not for RBMs. Indeed, when adding a new hidden unit, one should also update all the previous units $1 \rightarrow \mu - 1$, as the new hidden unit can perturb their statistics. Therefore, standard simultaneous optimization of all the hidden units is probably more effective than iterative optimization, but this formulation better highlights the individual roles of the hidden units and of the potential.

3.8 EXPLICIT FORMULA FOR SAMPLING AND TRAINING RBMS

We conclude this section with explicit formula for sampling and training RBMs. Due to the conditional independence property, sampling the conditional distributions is straightforward both for the visible and hidden layer; it requires sampling from $P(h_\mu|I) \propto e^{-\mathcal{U}_\mu(h_\mu)+Ih_\mu}$. Here, we give explicit formula for the average activity, transfer function $H(I) = \arg\max P(h|I)$ cumulant generative functions and its derivatives for the various potentials useful for sampling and training; in the following, we drop the hidden unit index μ.

3.8.1 *Bernoulli*

- $P(h|I) = \begin{cases} \frac{e^{g+I}}{1+e^{g+I}} & \text{if } h = 1 \\ \frac{1}{1+e^{g+I}} & \text{if } h = 0 \\ 0 & \text{Otherwise} \end{cases}$

- $\Gamma(I) = \log\left(1 + e^{g+I}\right)$
- $\langle h|I\rangle = \partial_I \Gamma(I) = \partial_g \Gamma(I) = \frac{1}{1+e^{-g-I}}$
- $\mathrm{Var}[h|I] = \partial_I^2 \Gamma(I) = \frac{e^{-g-I}}{(1+e^{-g-I})^2}$
- $H(I) = \mathbb{1}_{g+I\geq 0}$

3.8.2 *Potts*

- $P(z|I) = \frac{e^{g(z)+I(z)}}{\sum_{z'} e^{g(z)+I(z)}}$
- $\Gamma(I) = \log\left(\sum_z e^{g(z)+I(z)}\right)$
- $H(I) = \arg\max_{z'} g(z') + I(z')$

Note the degeneracy $g_i(z) \to g(z) + K_i$ and $w_{i\mu}(z) \to w_{i\mu}(z) + G_i$ for any K_i, G_i. In all experiments, we have removed the degeneracy by using the so-called zero-sum gauge: $\sum_z g_i(z) = 0$, $\sum_z w_{i\mu}(z) = 0\ \forall i, \mu$. SGD updates preserve the zero-sum gauge for the fields but not for the weights, see the gradient equations 3.11; the weights must be modified after each update to restore the zero-sum gauge: $w_{i\mu}(v) \to w_{i\mu}(v) - \frac{1}{q_v}\sum_{v'} w_{i\mu}(v')$, where q_v is the number of Potts states (e.g. 20 amino-acids) of visible units.

3.8.3 *Gaussian*

We write $\mathcal{N}(\mu, \sigma^2)$ the Gaussian distribution of mean μ and standard deviation σ. Then:

- $P(h|I) = \mathcal{N}\left(\frac{I-\theta}{\gamma}, \frac{1}{\gamma}\right)$
- $\Gamma(I) = \frac{(I-\theta)^2}{2\gamma} + \frac{1}{2}\log\frac{2\pi}{\gamma}$
- $\langle h|I\rangle = \partial_I \Gamma(I) = -\partial_\theta \Gamma(I) = \frac{I-\theta}{\gamma}$
- $\mathrm{Var}[h|I] = \frac{1}{\gamma}$
- $H(I) = \frac{I-\theta}{\gamma}$
- $\partial_\gamma \Gamma(I) = -\frac{1}{2}\langle h^2|I\rangle = -\frac{1}{2\gamma} - \frac{(I-\theta)^2}{2\gamma^2}$

3.8.4 *ReLU and dReLU*

As ReLU are special cases of dReLU (with $\gamma_- \to \infty$), we provide formula only for the latter potentials. We first introduce $\Phi(x) = \exp(\frac{x^2}{2}) \left[1 - \text{erf}(\frac{x}{\sqrt{2}})\right] \sqrt{\frac{\pi}{2}}$. Some useful properties of Φ are:

- $\Phi(x) \sim_{x \to -\infty} \exp(\frac{x^2}{2})\sqrt{2\pi}$
- $\Phi(x) \sim_{x \to \infty} \frac{1}{x} - \frac{1}{x^3} + \frac{3}{x^5} + \mathcal{O}(\frac{1}{x^7})$
- $\Phi'(x) = x\Phi(x) - 1$

To avoid numerical issues, Φ is computed in practice with its definition for $x < 5$ and with its asymptotic expansion otherwise. We also write $\mathcal{TN}(\mu, \sigma^2, \theta, +\infty)$ the truncated Gaussian distribution of mode μ, width σ and support $[\theta, +\infty]$. Then, we see first that $P(h|I)$ is equivalent to a mixture of two truncated Gaussians:

$$\begin{aligned} P(h|I) &= \begin{cases} \frac{1}{Z} \exp\left[-\frac{\gamma^+}{2}h^2 - (\theta^+ - I)\right] & \text{if } h \geq 0 \\ \frac{1}{Z} \exp\left[-\frac{\gamma^-}{2}h^2 - (\theta^- - I)\right] & \text{if } h \leq 0 \end{cases} \\ &= p^+ 1_{h \geq 0} \frac{e^{-\frac{\gamma_+}{2}h^2 + (I - \theta^+)h}}{Z_+} + p^- 1_{h<0} \frac{e^{-\frac{\gamma^-}{2}h^2 + (I-\theta^-)h}}{Z_-} \end{aligned} \tag{3.21}$$

Where $Z_\pm = \Phi\left(\frac{\mp(I-\theta^\pm)}{\sqrt{\gamma^\pm}}\right) \frac{1}{\sqrt{\gamma^\pm}}$, and $p_\pm = \frac{Z_\pm}{Z_+ + Z_-}$. We deduce the following formula:

- $P(h|I) = p^+ \mathcal{TN}\left(\frac{I-\theta^+}{\gamma_+}, \frac{1}{\gamma_+}, 0, +\infty\right) + p^- \mathcal{TN}\left(\frac{I-\theta^-}{\gamma^-}, \frac{1}{\gamma^-}, -\infty, 0\right)$
- $\Gamma(I) = \log\left[\frac{1}{\sqrt{\gamma^+}}\Phi\left(\frac{-I+\theta^+}{\sqrt{\gamma^+}}\right) + \frac{1}{\sqrt{\gamma^-}}\Phi\left(\frac{I-\theta^-}{\sqrt{\gamma^-}}\right)\right]$
- For $H(I)$ we distinguish two cases: the sparse case $\theta^+ > \theta^-$ (such as dReLU1 of Fig. 3.4,3.6), and the bimodal case $\theta^+ < \theta^-$ (dReLU2). For the former it writes:

$$H(x) = \text{ReLU}\left(\frac{I - \theta^+}{\gamma^+}\right) - \text{ReLU}\left(\frac{-I + \theta^-}{\gamma^-}\right)$$

Which justifies the name double ReLU 'dReLU'. Note the plateau betweeen $[\theta^-, \theta^+]$, which thresholds weak positive or negative inputs I and promotes sparse distributions. For the latter, it writes:

$$H(I) = \begin{cases} \frac{I-\theta^+}{\gamma^+} & \text{if} \quad I \geq \frac{\theta^+\sqrt{\gamma^-}+\theta^-\sqrt{\gamma^+}}{\sqrt{\gamma^+}+\sqrt{\gamma^-}} \\ \frac{I-\theta^-}{\gamma^-} & \text{if} \quad I \leq \frac{\theta^+\sqrt{\gamma^-}+\theta^-\sqrt{\gamma^+}}{\sqrt{\gamma^+}+\sqrt{\gamma^-}} \end{cases}$$

Note the discontinuity, which pushes an input I toward either strongly positive or negative values and promotes bimodal distribution. Both expressions are consistent for the equality case.

- $\langle h|I\rangle = p^+ \left[\frac{I-\theta^+}{\gamma^+} + \frac{1}{\sqrt{\gamma^+}\Phi\left(\frac{-I_+ + \theta^+}{\sqrt{\gamma^+}}\right)}\right] + p^- \left[\frac{I-\theta^-}{\gamma^-} - \frac{1}{\sqrt{\gamma^-}\Phi\left(\frac{I-\theta^-}{\sqrt{\gamma^-}}\right)}\right]$
- $\mathrm{Var}[h|I] = \frac{p^+}{\gamma^+} + \frac{p^-}{\gamma^-} + p^+p^- \left(I\left(\frac{1}{\gamma^+} - \frac{1}{\gamma^-}\right) - \frac{\theta^+}{\gamma^+} + \frac{\theta^-}{\gamma^-} + \frac{1}{\sqrt{\gamma^+}\Phi\left(\frac{-I+\theta^+}{\sqrt{\gamma^+}}\right)} + \frac{1}{\sqrt{\gamma^-}\Phi\left(\frac{I-\theta^-}{\sqrt{\gamma^-}}\right)}\right)$
$$\frac{I\left(\frac{1}{\gamma^+}-\frac{1}{\gamma^-}\right) - \frac{\theta^+}{\gamma^+} + \frac{\theta^-}{\gamma^-} - \frac{1}{\sqrt{\gamma^+}\Phi\left(\frac{-I+\theta^+}{\sqrt{\gamma^+}}\right)} + \frac{1}{\sqrt{\gamma^-}\Phi\left(\frac{I-\theta^-}{\sqrt{\gamma^-}}\right)}}{\frac{\Phi\left(\frac{-I+\theta^+}{\sqrt{\gamma^-}}\right)}{\sqrt{\gamma^+}} + \frac{\Phi\left(\frac{I-\theta^-}{\sqrt{\gamma^-}}\right)}{\sqrt{\gamma^-}}}$$
- $\partial_{\theta^+}\Gamma(I) = -\langle \max(h,0)|I\rangle = -p^+ \left[\frac{I-\theta^+}{\gamma^+} + \frac{1}{\sqrt{\gamma^+}\Phi\left(\frac{-I_+ + \theta^+}{\sqrt{\gamma^+}}\right)}\right]$
- $\partial_{\theta^-}\Gamma(I) = -\langle \min(h,0)|I\rangle = -p^- \left[\frac{I-\theta^-}{\gamma^-} - \frac{1}{\sqrt{\gamma^-}\Phi\left(\frac{I-\theta^-}{\sqrt{\gamma^-}}\right)}\right]$
- $\partial_{\gamma^+}\Gamma(I) = -\frac{1}{2}\langle \max(h,0)^2|I\rangle = -\frac{1}{2}p^+ \left[\frac{1}{\gamma^+} + \left(\frac{I-\theta^+}{\gamma^+}\right)^2 + \frac{I-\theta^+}{\gamma^+\Phi\left(\frac{-I+\theta^+}{\sqrt{\gamma^+}}\right)}\right]$
- $\partial_{\gamma^-}\Gamma(I) = -\frac{1}{2}\langle \min(h,0)^2|I\rangle = -\frac{1}{2}p^- \left[\frac{1}{\gamma^-} + \left(\frac{I-\theta^-}{\gamma^-}\right)^2 - \frac{I-\theta^-}{\gamma^-\Phi\left(\frac{I-\theta^-}{\sqrt{\gamma^-}}\right)}\right]$

Part II

LEARNING ALGORITHMS FOR BOLTZMANN MACHINES AND RESTRICTED BOLTZMANN MACHINES

This short part is dedicated to maximum likelihood training in BM and RBM. As shown in Section 3.4, both models require evaluation of the moments of the distribution, which is NP hard in the general case. Moreover, in the case of RBM with continuous units, ill-conditioning of the hessian complexifies training. Overall, feature extraction with RBM is fairly robust but reaching good generative performance can be challenging, and numerous articles have studied the subject. We will briefly review them, before presenting two developments introduced for the purpose of this thesis: a new sampling algorithm and a new reparameterization trick.

4

THE MOMENT EVALUATION PROBLEM

4.1 BACKGROUND

We recall that the models are fitted through likelihood optimization, which can be carried out by a gradient ascent algorithm, consisting of successive updates of the form $\theta^{(t+1)} = \theta^{(t)} + \mathrm{lr}_t \nabla_\cdot \mathcal{L}$, where lr_t is the learning rate at time t, until convergence is reached. For BM and RBM, the likelihood gradient takes the form of a difference between a data average and model average:

$$\nabla_\theta \mathcal{L} = -\left\langle \nabla_\theta E(\mathbf{v}, \theta) \right\rangle_d + \left\langle \nabla_\theta E(\mathbf{v}, \theta) \right\rangle_m \tag{4.1}$$

A gradient update therefore consists in simultaneously pushing down the energy of the data configuration and pushing up the energy of the current model distribution; convergence is reached once the two effects compensate exactly. For BM and RBM, the left hand term can be easily evaluated from the data. For BM, $\frac{\partial E(\mathbf{v})}{\partial g_i} = v_i$ and $-\frac{\partial E(\mathbf{v})}{\partial J_{ij}} = v_i v_j$, such that after averaging, we obtain exactly the first and second order moments $f_i = \langle v_i \rangle_d, f_{ij} = \langle v_i v_j \rangle_d$; they can be computed once before the training starts. For RBM, the derivatives are non-linear moments and depend on the current parameter estimates, see Eqn. (3.11); they must be evaluated after each parameter update. For speed purposes, we evaluate the data average using only a small mini-batch of data ($N_{\text{batch}} \sim 100$ in most of our experiments), and iterate multiple time over all the mini-batches. Provided that the gradient can be evaluated and that the learning rate rate slowly decays to zero, this optimization method, termed Stochastic Gradient Descent (SGD) correctly converges toward a local maximum of the likelihood. Moreover, it exhibits increased speed and better behavior for non-convex optimization, see [98] for more information.

On the other hand, the right hand term is hard to evaluate, since neither analytical evaluation (cost is exponential in N) nor direct sampling from P are possible (i.e. when samples from P can be obtained by transforming of uniformly distributed samples). In their original formulation of BM/RBM, Ackley, Hinton and Sejnowski used Markov Chain Monte Carlo (MCMC) to

simulate the model. We recall briefly that MCMC consists in constructing a Markov chain such that the desired distribution (here P) matches the Markov chain distribution of samples in the limit of an infinite number of Markov steps, see [99] for an introduction. The greater the number of steps, the closer the Markov chain distribution is to the desired distribution. In our context, at each step of the gradient descent, we launch a set of N_{chains} Markov Chains, wait until convergence, then evaluate the moments from the samples obtained. Though this is possible in principle, the very long computational time presents a major difficulty for doing so. For instance, a naive Metropolis-Hasting or Gibbs MCMC of a Curie-Weiss model, i.e. RBM with N ± 1 visible units, $M = 1$ hidden unit, and uniform weight matrix $w_{i1} = w/N$) requires of the order of $\exp(Nw^2)$ steps to converge to the equilibrium distribution when $w > 1$. More generally, naive MCMC generally fails whenever the regions of the configuration space with low-energy do not form a connected space, in the sense that one must transit through (very) high-energy configurations to go from one region to the other; transitions are therefore extremely rare, and convergence to the equilibrium distribution is never observed in practice for large N. The original experiments of Ackley et al. were therefore limited to toy data sets, and BM/RBM rapidly lost traction in favor of less computationally heavy methods such as backpropagation. Since then, numerous heuristics were developed for handling this problem and are briefly presented here.

4.1.1 *Contrastive Divergence*

Contrastive Divergence (CD) is a MCMC based method introduced by Hinton in 2002 for training RBM [70]. It is a simplification of the original MCMC sampling algorithm in which instead of starting Markov chains from random configurations and waiting until equilibrium is reached, each chain is initialized with a data sample, and only a few N_{MC} Gibbs sampling steps are applied before evaluating the gradient. In practice, we set $N_{\text{chains}} = N_{\text{batch}}$, and for a given SGD step, we use the same data samples for evaluating the data average of Eqn. (4.1) and for initializing the MCMC chains that will be used; the gradient therefore quantifies a 'contrast' between the initial data and the chains that have 'diverged' away from them. The intuition behind CD is that if we have $P \sim P_d$, the Gibbs sampling leaves both the model and data distribution invariant, such that the gradient vanishes. Conversely, if we update the parameters such that the Gibbs sampling step(s) leave invariant the data distribution, we should bring P close to P_d. For instance, for a data set constituted by two handwritten digits (a 0 and a 1), CD learns by 'digging' the energy landscape around the two original samples, see Fig. (4.1). More formally, Bengio and Delalleau later constructed a formal

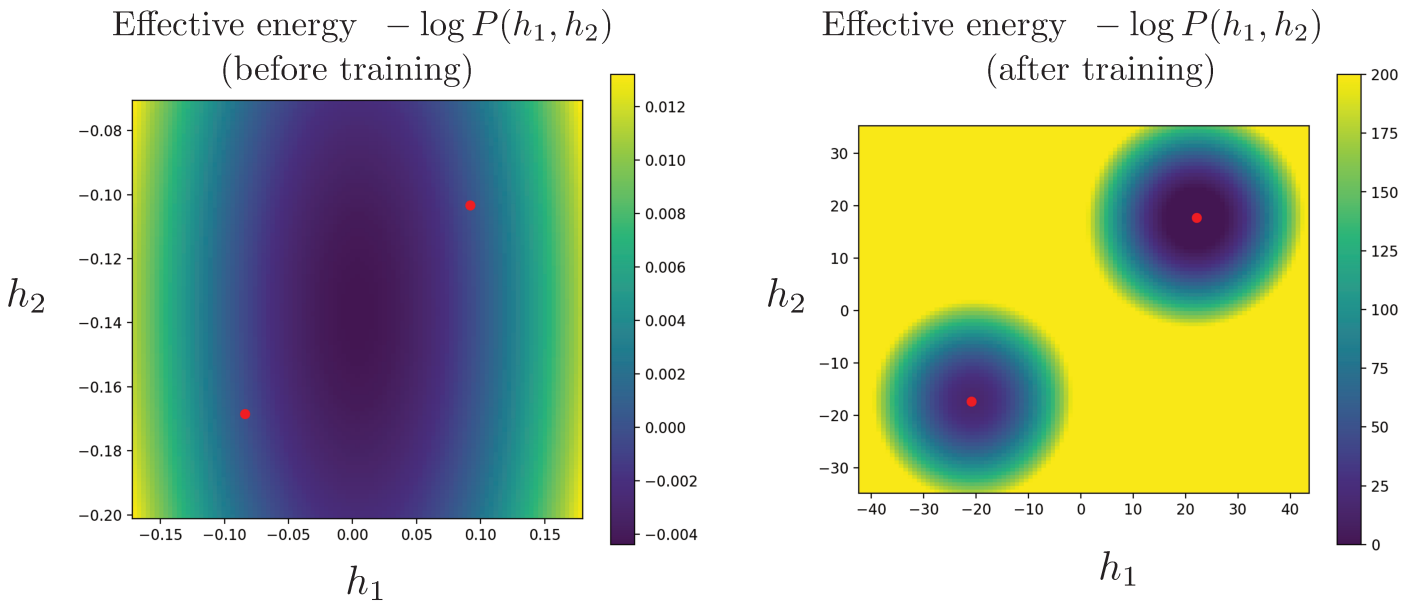

Figure 4.1: CD learning on a data set constituted by two digits. Energy landscape in hidden layer representation $P(h)$ before (left) and after (right) training; red dots denote the two data samples. Here, CD successfully digs attractors around the two samples

series expansion of the log-likelihood via Gibbs updates, and showed that CD learning is equivalent to a gradient descent over the truncated log-likelihood expansion [100]. Compared to the original Monte Carlo learning algorithm, CD is biased but much faster: it avoids the long 'burn-in' time necessary for Markov Chains to reach high-probability configurations, and prevents them from being stuck in a mode of the model, as they are each time initialized in a different mode. Using CD, Hinton presented the first successful training of RBM on a data set of honorable size, namely the MNIST handwritten digits data set. Though strokes-like features can be found, the samples generated by the model are not very good. This is because the Monte Carlo sampling explores only a neighborhood of the data samples, such that regions far away from the original data are never seen. This is problematic because learning some samples can increase the probability of other configurations away from them; think for instance of the Hopfield model: learning M patterns can generate high-probability spurious states. In Fig. (4.2), we show a similar example with three digits instead of two. CD learning fails to get rid of spurious states because samples (green dots) cannot escape from the local attractor, even after 500 Gibbs steps. As for PCD (see below) CD learning can lead to divergence of the likelihood [75, 101, 102].

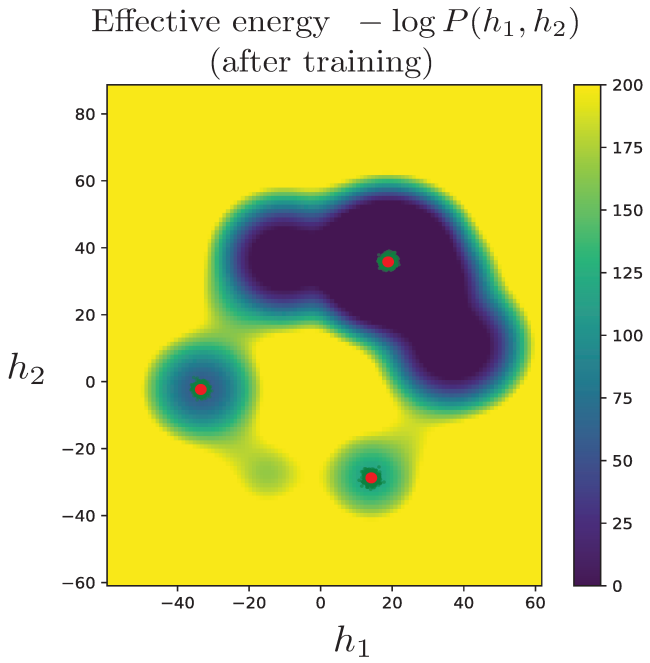

Figure 4.2: Same with three digits. Green dots indicate CD samples with 500 steps, starting from the data instances; they cannot reach the low-energy spurious attractors, though they have low energy as well

4.1.2 *Persistent Contrastive Divergence*

Instead of initializing the Markov Chains from samples at each gradient update, Tieleman and Hinton later proposed to maintain the same set of (persistent) Markov chains from one gradient update to the other, performing only a few N_{MC} steps between each update [103]. The intuition behind Persitent Contrastive Divergence (PCD) is that provided that the samples at a given time are effectively at equilibrium and the probability distribution varies slowly from one update to the other, then only a few steps should be required to adjust to the new equilibrium distribution. Indeed, it can be shown that when the learning rate tends to zero, PCD gives exactly samples from the probability distribution [104]. In practice, due to slow mixing rates, samples from PCD are often stuck in one mode, and the Markov Chains are unable to track global evolutions of P_m, such as the apparition of spurious modes, or the relative probabilities of each mode. Consider for instance a trivial data set with $N = 2$ Bernoulli units, and $p_{00} = p_{11} = 0.48$, $p_{01} = p_{10} = 0.02$. We train a RBM with a single Bernoulli hidden unit, using either exact moment evaluation (there are only 8 states) or PCD (with 1 or 2 Gibbs steps between each update), using a fixed learning rate $\mathrm{lr} = 0.1$, and $N_{\text{batch}} = N_{\text{chains}} = 10$. The exact method quickly converges to the optimum, whereas PCD learning leads to divergence

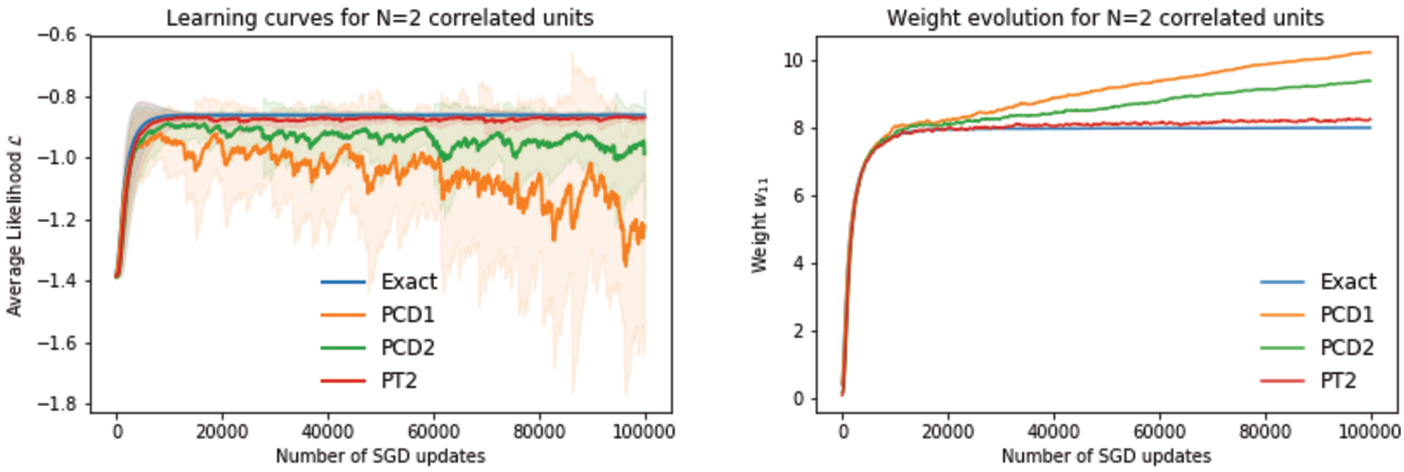

Figure 4.3: PCD learning on a $N = 2$ toy data set, using a RBM with $M = 1$. Left: Evolution of the train set likelihood (exact computation). Full line denote the exponential smoothing of $\mathcal{L}(t)$ and error indicate fluctuations around the smoothing; Right: Evolution of one of the two weights w_{11}. PCD learning results in divergence, even for a trivial case. Parallel Tempering (PT, see below) with 2 replica (at $\beta = 0, 1$) correctly converges to the optimum

of the likelihood and the weights, see Fig. (4.3). Indeed, since transitions 00 to 11 are very rare (one must first get $h = 1$ from 00), the persistent chains cannot track well the relative changes in p_{00},p_{11}. For instance, if we start with a distribution having an excess of 11, the gradient update will quickly increase the probability 00 but the samples will take a lot more time to adjust; such that the excess remains while learning continues, resulting in an overshoot with a very high probability of 00. Though this divergence phenomenon can be reduced by using smaller learning rate and more updates, the mixing time typically increases as the model gets better (and the energy landscape steeper), such that PCD generally results in divergence of the likelihood [75, 101, 102]. In practice, we found that PCD can converge reasonably well provided that sufficient Monte Carlo updates are used and that we gradually anneal the learning rate to zero, so as to avoid divergence. Starting from an initial value lr_i, a geometric decay of learning rate starts after da% (e.g. 50%) of the training until a final value lr_f (e.g. $10^{-3} \times \mathrm{lr}_i$). Results may however vary a lot on the training time: too few updates can result in underfit, whereas too many updates can lead to divergence.

4.1.3 *Parallel Tempering*

In an attempt to improve the efficiency of MCMC simulations, non-local Monte Carlo algorithms from the physics/chemistry literature were proposed instead of traditional Gibbs sampling for PCD learning [102,105,106]. One example is the replica exchange method *i.e.* parallel tempering [107]. Parallel Tempering (PT) consists in simulating in parallel several replica of the system $\{(\mathbf{v_r}, \mathbf{h_r}), r = 1..N_R\}$ each drawn from the Gibbs distribution at inverse temperature $\beta_r \in [0, 1]$:

$$P_{\beta_r}(\mathbf{v}, \mathbf{h}) = \frac{1}{Z_{\beta_r}} \exp\left[-\beta_r E(\mathbf{v}, \mathbf{h}) - (1 - \beta_r) E_0(\mathbf{v}, \mathbf{h})\right] \tag{4.2}$$

Where the β_r are ordered between 0 and 1 (the original distribution), with $\beta_1 = 0$, $\beta_{N_R} = 1$, and where $E_0 = \sum_i \mathcal{U}_i^0(v_i) \sum_\mu \mathcal{U}_\mu^0(h_\mu)$ has no interaction terms, such that P_0 mixes instantaneously. The set of intermediate distributions effectively interpolates between the fast mixing P_0 and the target P_1. During sampling, we propose configurations swaps between P_{β_r} and $P_{\beta_{r+1}}$: $\{\mathbf{v}_r, \mathbf{v}_{r+1}\} \leftarrow \{\mathbf{v}_{r+1}, \mathbf{v}_r\}$ which are accepted with the Metropolis acceptance rate:

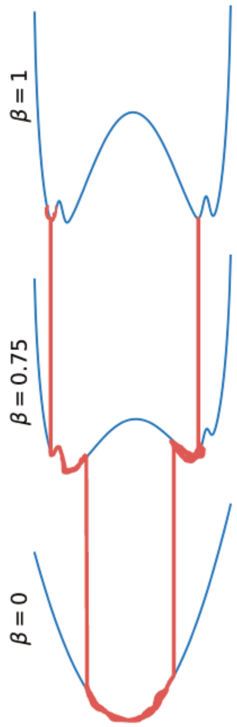

Figure 4.4: Principle of Parallel Tempering (PT)

$$AR_r = \min\left\{1, \frac{P_{\beta_r}(\mathbf{v_{r+1}}) P_{\beta_{r+1}}(\mathbf{v_r})}{P_{\beta_r}(\mathbf{v_r}) P_{\beta_{r+1}}(\mathbf{v_{r+1}})}\right\} \tag{4.3}$$

Where $P_\beta(\mathbf{v})$ is the marginal of distribution 4.2 [1]. The intuition behind PT is illustrated in Fig. 4.4: A particle trapped in a local energy minimum at $\beta = 1$ is moved to higher temperature, where the energy barrier is lower and it can diffuse freely; once back at the original temperature, it has effectively escaped to another mode. For RBM trained on MNIST, this allows particles to switch significantly faster from one digit type to the other. In practice, exchanges are proposed after a Gibbs update of all replica chains, alternatively for all even r or for all odd r and P_0 is chosen as the closest independent model to the data

1 Technically, we perform here a replica exchange on the marginal $P_\beta(\mathbf{v}) = \int \prod dh_\mu P_\beta(\mathbf{v}, \mathbf{h})$ rather than on the joint probability distribution $P_\beta(\mathbf{v}, \mathbf{h})$. Both are possible but the former is better as the system is about twice as small, resulting in significantly higher acceptance rates. The downside is that hidden units must be resampled after swaps

(ie that minimizes $D_{KL}(P_d|P_0)$). Best results are obtained when the $\{\beta_r\}$ are not evenly spaced, as P_β can change slowly or abruptly with β, e.g. at phase transitions; we present our approach for dynamically adjusting the β in the next section.

PT sampling-based training with two replica of the system at $\beta = 0$ and $\beta = 1$ gracefully solves the mixing problem and subsequent divergence problem in the toy data set presented above for PCD, see Fig. 4.3. Indeed, the system at $\beta = 0$ samples at each step visible layer configurations 00,01,10,11 with equal probability $\frac{1}{4}$; therefore, a Markov Chain at $\beta = 1$ stuck in 00 may swap its configuration its configuration with a 11 at any time. The acceptance rates of Eqn. 4.3 allow to maintain balance between the 4 possibles configurations: configurations $01, 10$ at $\beta = 0$ are less likely to be swapped to $\beta = 1$ than 00,11. PT training was shown to be superior to regular CD/PCD training in terms of likelihood improvements in numerous cases [102, 105, 106]. In practice, although accepting moves with Metropolis rates guarantees unbiased samples, the acceptance rates can be very low when P_1 and P_0 are very different. In that case, large number of replica N_R (10 to 40) are required to obtain significant acceptance rates, even for intermediate values of $N \sim 10^{2-3}$. Moreover, the above intuitive picture can become incorrect when entropic effects are taken into consideration: the particles may encounter entropic barriers at $\beta = 0$ that break ergodicity. Indeed, if one of the modes has lower energy but higher diversity than the others (e.g. a mixture of two 1D gaussians with $\sigma_1 >> \sigma_2$), it will completely dominate at low, non-zero β, and there will be no way for a particle lying in other modes at $\beta = 0$ to climb and reach $\beta = 1$. In physicist's terms, Parallel Tempering does not work whenever a system undergoes a first order phase transition. For instance, consider a simple binary data set drawn from the following Mixture of Independent distribution:

$$P(\mathbf{v}) = \frac{1}{3}\sum_{k=1}^{3} \frac{e^{g_k \sum_{i=1}^{N} v_i}}{(1 + e^{g_k})^N} \tag{4.4}$$

Where $g_k = \log \frac{\mu_k}{1-\mu_k}$, and $\mu_1 = 0.1$, $\mu_2 = 0.5$, $\mu_3 = 0.9$. It is a trimodal distribution, with modes differentiated by their average activity $m = \frac{1}{N}\sum_{i=1}^{N} v_i$. The induced distribution $P(m)$ has three peaks at μ_1, μ_2, μ_3 of identical area, see Fig. (4.5 b). We now argue that an RBM trained to reproduce $P(\mathbf{v})$ cannot be sampled from using Gibbs or PT MCMC sampling. Clearly, whenever N is large, no transitions between the three modes are observed and Gibbs sampling

fails. For PT, since $\langle v_i \rangle = 0.5$, we have $E_0 = 0$, such that the intermediate distribution is simply:

$$P_\beta(\mathbf{v}) \propto \left(\sum_{k=1}^{3} \frac{e^{g_k \sum_{i=1}^{N} v_i}}{(1+e^{g_k})^N} \right)^\beta \tag{4.5}$$

One may expect that as the energy landscape is flat at $\beta = 0$, particles should be able to diffuse freely and PT should thermalize. This is not the case, as they will actually encounter an entropic barrier. Indeed, summing over all v_i, we get for large N:

$$P_\beta(m) \propto \exp\left[\beta \log\left(\sum_k e^{N m g_k} \right) + N\left(m \log(m) + (1-m)\log(1-m) \right) \right] \tag{4.6}$$

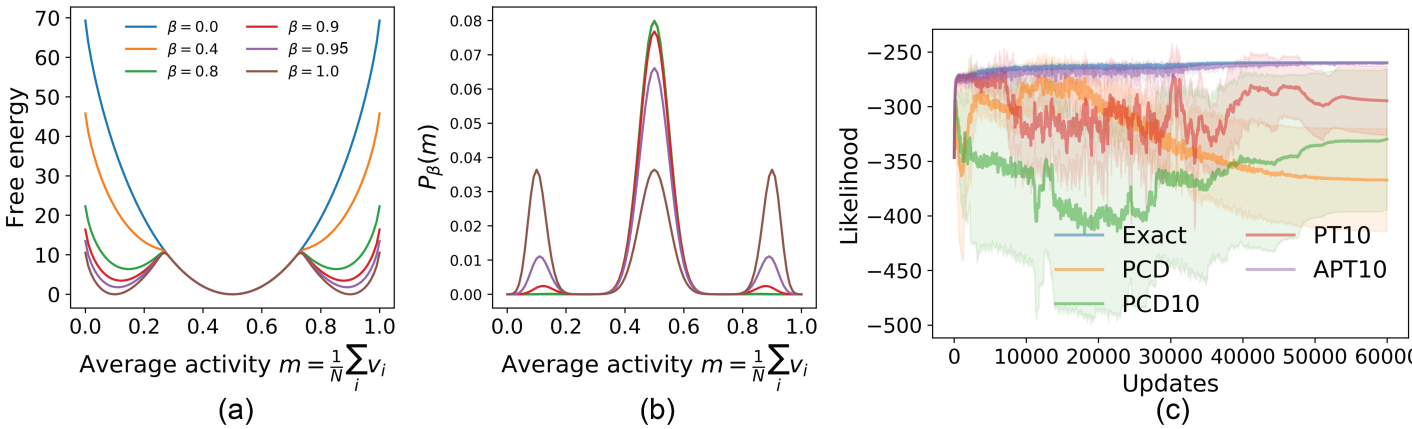

Figure 4.5: **Counter example for PCD and PT learning**. (a) Free energy $F(m) \sim -\log P_\beta(m)$ of the average activity m, for various β (b) Corresponding probability distribution. Crucially, the peaks at $m = 0.1$, $m = 0.9$ already exist at intermediate β, but are completely dominated by the central, high-entropy mode $m = 0.5$. When $\beta \to 1$, the distribution becomes brutally multimodal, and at no point intermediate values $m \sim 0.3, 0.7$ are likely. (c) Learning P with an RBM with two Bernoulli hidden units, for $N = 500$. Both PCD, PCD10 (= 10 Gibbs updates) and PT10 (= 10 replica) fail, whereas APT10 (see next section) correctly matches the performance of exact ML training.

The free energy (i.e. $-\log P_\beta(m)$) and the corresponding distribution are shown for various β in Fig. (4.5 a). Clearly, Parallel Tempering will not work,

as for $\beta < 1$ the overwhelming majority of configurations have $m \sim 0.5$ and the two other peaks are not populated. Moreover, even at $\beta \sim 1 - \frac{\epsilon}{N}$ where all peaks are populated the intermediate regions $\mu_2 < m < \mu_3$, $\mu_1 < m < \mu_2$ are never populated, such that there are no path from one mode to the other. In practice, we indeed observe that training with Gibbs and PT fails, whereas Augmented Parallel Tempering (APT, introduced below) trivially works, as $P_0 = P_1$, see panel c.

This counter-example may be ubiquitous in real data sets. First, whenever a mode has higher entropy than the others (such as mode 2 here), we expect it to dominate at low β. In MNIST, we have observed for instance that the set of digits 1 has much lower entropy that the others, such that they are unseen at low β. More broadly, whenever the system undergoes a first-order phase transition, *i.e.* when β essentially changes the relative proportion of each mode but not their location, we expect PT to fail.

4.1.4 *Methods not based on sampling*

Finally, we briefly mention training methods that are not based on sampling. The moments can be evaluated using analytical approximations, such as the mean-field approximation [108], loopy belief propagation, TAP expansion, see [32] for a review. For instance, for BM with binary units taking values ± 1, the first and second moments are, in the mean-field approximation:

$$\begin{aligned} C_{ij} &\equiv f_{ij} - f_i f_j = -J^{-1}_{ij} \\ f_i &= \tanh\left(h_i + \sum_j J_{ij} f_j\right) \end{aligned} \tag{4.7}$$

At fixed f_i,f_{ij}, these equations can be inverted to obtain directly an estimate for the fields and couplings. In the case of BM, mean-field like approximations often give reasonable estimates for the interaction matrix, but they are in general not generative, *i.e.* sampling from the distribution does not reproduce the moments from the data. Indeed, these approaches are justified only in the limit of weak interactions or of tree-like graphs of interactions. More recently, Cocco and Monasson proposed an Adaptive Cluster Expansion (ACE) for computing the moments [82, 109, 110], that is justified when the interaction graph is sparse. Though the computational cost is much heavier than mean-field

or TAP approximations, ACE can correctly reproduce the data distribution, which is crucial for generative or scoring purposes.

In the case of RBM, the mean-field equations can be solved by fixed point iteration and be used to compute the gradient [111]; Tielemann and Hinton however showed that they produce neither meaningful features nor digits. More recently, Gabrie et al. showed that Thouless-Anderson-Palmer (TAP) expansion could retrieve meaningful features, and that the solutions of the TAP fixed point equations are reminiscent of original data samples [112, 113]. TAP-based learning algorithms could be interesting for training RBM on proteins, as strong regularization is often used in that case; such that interactions are relatively weak. Generalization of ACE to RBM could be interesting as well for this purpose.

BM and RBM can also be trained using different objectives than maximum likelihood, such as pseudo-likelihood maximization (PLM) or minimum probability flow [114]. In particular, BM trained by PLM were found very successful in the context of structure predictions in proteins [115].

4.2 AUGMENTED PARALLEL TEMPERING

4.2.1 *Principle*

In an attempt to overcome the issues found in CD, PCD or PT sampling, we propose a new sampling algorithm, Augmented Parallel Tempering. It shares benefits with both CD and PT learning. As a starting point, we recall first that although Parallel Tempering is well suited for exploring an unknown energy landscape, our sampling problem is substantially easier because we already know where to look: samples should be located close to the data. We would like to use this available information very much like Contrastive Divergence, but within an unbiased framework. To this end, we propose to learn P_0 from data using a Mixture of Independent model (MoI):

$$P_0(\mathbf{v}) = \sum_{z=1}^{Z} \prod_i P_0(v_i|z) P_0(z) \tag{4.8}$$

MoI are directed graphical models, can learn multimodal distribution and are relatively simple to fit to data using the Expectation-Maximization algorithm, combined with some tricks such as KM++ initialization [116] and Split-Merge

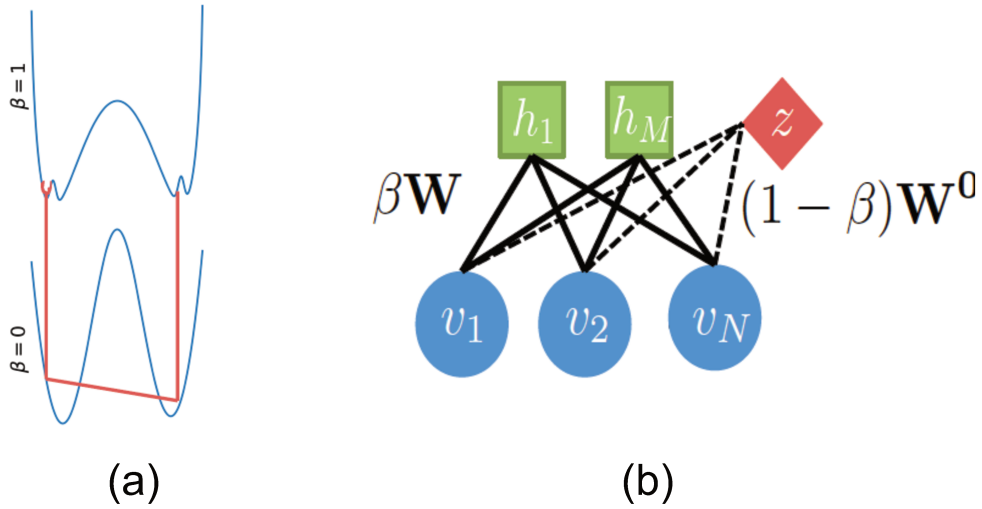

Figure 4.6: (a) Schematic view of a trajectory in Augmented Parallel Tempering (b) Augmented Architecture interpolating between the MoI ($\beta = 0$) and the RBM ($\beta = 1$)

heuristics [117]. Typically, each mixture corresponds to one of the different modes of the data, e.g. different digit types, see a visualization of the conditional probability $P(v_i = 1|z)$ in Fig. 2.2b. Crucially, MoI allow direct sampling, such that a Markov Chain on P_0, defined as successive direct samples is both ergodic and multimodal. In principle, using MoI rather than an independent model would replace long, connex trajectories in the $\beta \times$ Configuration space with shorter, non-local ones, see Fig.4.6a. Moreover, P_0 can be significantly closer to P_1, thus reducing the number of replica required to achieve reasonable acceptance rates. Overall, we expect this modification to overcome, at reasonable extra cost, the most obvious limitation of MCMC-based learning, namely its inability to fit multimodal, non-connected data distributions. In the following, we focus on an implementation for binary visible units RBM, but a similar algorithm can be derived for other RBM or BM.

4.2.2 *Implementation*

Choice of interpolating distributions

Like standard PT, P_0 and P_1 can be embedded in a single parametric family; to derive P_β for intermediate β, we observe first that an MoI parameterized by

$P_0(z)$, $P_0(v_i|z)$ is equivalent to a RBM with a single categorical hidden unit z with parameters:

$$\begin{aligned} g_i^0 &= \log\left(\frac{\langle v_i\rangle_d}{1-\langle v_i\rangle_d}\right) \\ w_i^{MoI}(z) &= \log\left(\frac{P(v_i=1|z)}{1-P(v_i=1|z)}\right) - g_i^0 \\ g_z(z,0) &= \log P_0(z) - \sum_i \log\left[P(v_i=1|z)(1-P(v_i=1|z))\right] \end{aligned} \tag{4.9}$$

Where we chose arbitrarily, but without loss of generality the g_i^0 as in PT. Then, we define the augmented RBM architecture shown in Fig.4.6b: one visible layer, the RBM hidden layer and the mixture model node. At intermediate β, its distribution writes:

$$\begin{aligned} P_\beta(\mathbf{v},\mathbf{h},z) = \frac{1}{Z_\beta}\exp\Bigg(&\sum_i \beta g_i + (1-\beta)g_i^0 + \sum_\mu -\beta\mathcal{U}_\mu(h_\mu) - (1-\beta)\mathcal{U}_\mu^0(h_\mu) \\ &+ g_z(z,\beta) + \beta\mathbf{h}^T\mathbf{W}\mathbf{v} + (1-\beta)z^T\mathbf{W}^{\mathbf{MoI}}\mathbf{v}\Bigg) \end{aligned} \tag{4.10}$$

Where U_μ^0 are chosen as in PT and $g_z(z,1) = \log P_0(z)$, such that $P_1(z) = P_0(z)$. By construction, the marginal distributions $P_1(v)$ and $P_0(v)$ match respectively the RBM and MoI marginal probabilities. Crucially, $g_z(z,\beta)$ is not necessarily linear, because the marginal $P_\beta(z)$, and thus the probabilities of each mode would not be preserved. For instance, if $\mathbf{W} = 0$, the linear interpolation $g_z(z,\beta) = g_z(z,0) + \beta\left[g_z(z,1) - g_z(z,0)\right]$ gives $P_\beta(z) \propto P_0(z)\prod_i \left(P(v_i=1|z)^\beta(1-P(v_i=1|z)^{1-\beta}\right)$, which can brutally deviate from its initial/final value. In fact, some modes z can be completely erased at intermediate β, see Fig. (4.7). Therefore, a particle lying in z cannot be swapped to β, and is trapped either below or above β, resulting in poor ergodicity and slow mixing. Instead, we use a polynomial interpolation of the form:

$$g_z(z,\beta) = g_z(z,\beta) = g_z(z,0) + \beta\left[g_z(z,1) - g_z(z,0)\right] + \sum_{k=0}^{D}\beta(1-\beta)(2\beta-1)^k C(k) g_z^{(k)}(z) \tag{4.11}$$

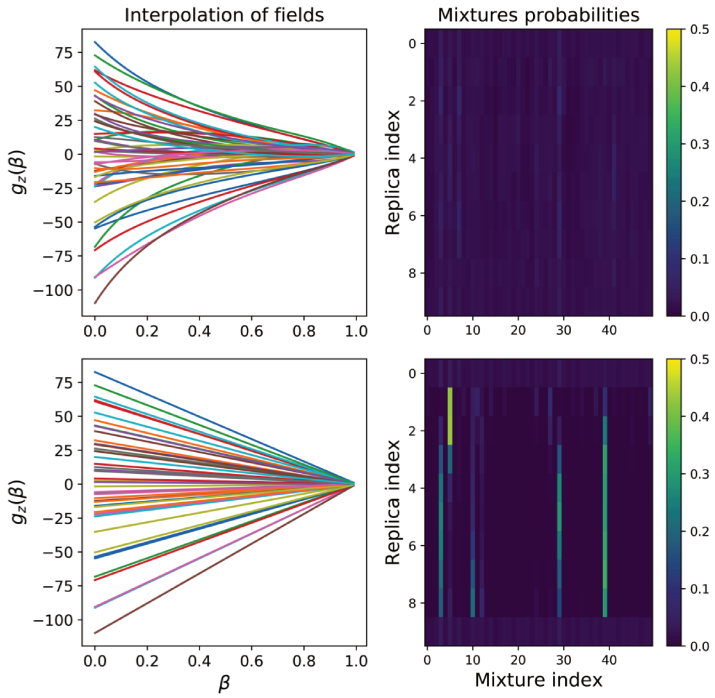

Figure 4.7: **Repartition of samples into modes** For a dReLU RBM trained on MNIST with 400 hidden units and a MoI with 50 modes, $P_\beta(z)$ for $N_R = 10$ using a linear interpolation (top) and non-linear interpolation (bottom) of the fields $g_z(z, \beta)$. Using non-linear interpolation allows to maintain a balance between all modes, increasing ergodicity.

Where the normalization factor $C(k) = \frac{k+2}{2}\left(\frac{k+2}{k}\right)^{\frac{k}{2}}$ ensures that each additional term reaches a maximum 1 for $\beta \in [0, 1]$, and the degree D is 3 by default. We dynamically adjust the additional variables $g_z(k)(z)$ such that the marginals $P_{\beta_r}(z)$ remain as close as possible to $P_0(z)$ in terms of KL divergence:

$$g_z^{(k)}(z) = \arg\min \sum_{r,z} P_0(z) \log\left(\frac{P_0(z)}{P_{\beta_r}(z)}\right) \tag{4.12}$$

In practice, the minimization is carried out by Newton method, with a block diagonal hessian approximation (each z corresponds to a block). Let $M_{rk} = C(k)\beta_r(1-\beta_r)(2\beta_r - 1)^k$ be the matrix of coefficients, M^+ its pseudo-inverse (with default conditioning 0.1). Writing in vector notation $\mathbf{g_z}(\mathbf{z}) = \{g_z^{(0)}(z), ..., g_z^{(D)}(z)\}$ and $\mathbf{P}(\mathbf{z}) = \{P_{\beta_2}(z), ..., P_{\beta_{N_R-1}}(z)\}$ the observed probabili-

ties for the mini-batch of Markov chains and $\odot$ the elementwise product, we obtain the following update rule for $\mathbf{g}_{\mathbf{z}}(\mathbf{z})$:

$$\mathbf{g}_{\mathbf{z}}(\mathbf{z}) \to \mathbf{g}_{\mathbf{z}}(\mathbf{z}) + \rho(M^+)^T \left\{ [\mathbf{P}(\mathbf{z})(\mathbf{1} - \mathbf{P}(\mathbf{z})) + \epsilon \mathbf{1}]^{\odot -1} \odot M^+ \left(\mathbf{1} P_0(z) - \mathbf{P}(\mathbf{z}) \right) \right\} \tag{4.13}$$

Where ϵ is set to $\frac{1}{N_{chains}}$ to avoid divergence due to finite sampling, and the learning rate ρ is initially set to 0.05, and decays exponentially throughout learning. Example of fields inferred are shown in Fig. (4.7); the non-linear interpolation correctly maintains balance between modes at all sites.

Choice of inverse temperatures

A common rule of thumb for choosing $\{\beta_r\}$ in physical simulations is to set β such that the average acceptance rates $\langle AR_r \rangle$ are approximately uniform across all pairs of replica. To this end, the following heuristic works well in practice and adds limited computational overhead. We define a set of 'springs stiffness' between pairs of replicas as, and their associated 'elastic energy' as:

$$\begin{aligned} K_r &= \max \left[1 - \frac{\langle AR_r \rangle}{\frac{1}{N_R - 1} \sum_{r'} \langle AR_{r'} \rangle}, 0 \right] \\ \mathcal{E}(\{\beta_r\}) &= \frac{1}{2} \sum_{r=0}^{N_R - 1} K_r (\beta_r - \beta_{r+1})^2 \end{aligned} \tag{4.14}$$

K_r are zero if the swap $(r, r+1)$ have larger acceptance rate than average, and positive elsewhere. We then update the β_r so as to minimize the 'elastic energy', with the boundary conditions $\beta_1 = 0, \beta_{N_R} = 1$. Intuitively, it moves closer pairs of inverse temperature (β_r, β_{r+1}) that have low swap acceptance rates, thus regressing it to the mean value. The average acceptance rates are computed using the current mini-batch and exponentially smoothed, and the update writes: $\beta^{(t+1)} = \mathrm{lr}_\beta \arg\min \mathcal{E} + (1 - \mathrm{lr}_\beta)\beta^{(t)}$. Lastly, the total number of replica N_R is adjusted dynamically during training so as to maintain high average swap acceptance rates. Starting from $N_R = 2$ at the beginning of training, new replica are spawned so as to maintain an average acceptance rates above some target value, e.g. 0.3. The new replica is added at inverse temperature 0 and its Markov chains are initialized with the previous $\beta = 0$ samples. Additionally, to anticipate the subsequent readjustment of the β's, we

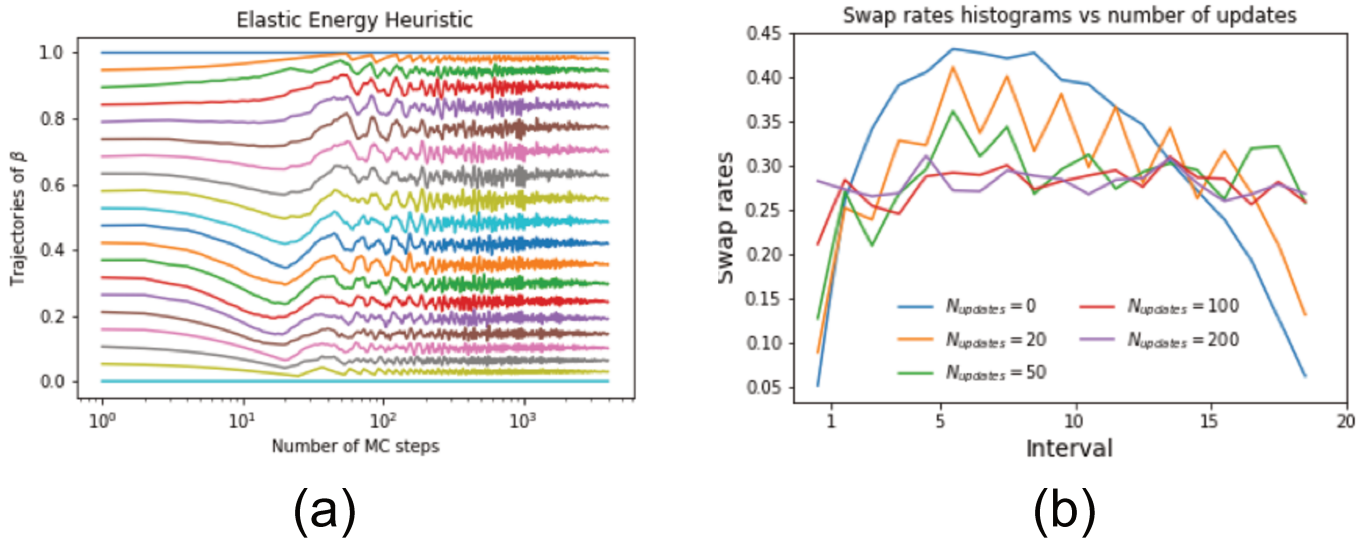

Figure 4.8: **Adapting the set of inverse temperatures** For a RBM with dReLU hidden units trained on MNIST, we perform a pure PT sampling phase while adapting the set of temperatures. Left: Trajectories of the temperatures (log scale). Right: Swap rates profiles at various number of Gibbs steps.

slightly push up the inverse temperature of the previous Markov chains, with $\beta^{(t+1)} = \frac{N_R-2}{N_R-1}\beta^{(t)} + \frac{1}{N_R-1}$ (the $\beta = 1$ chain is not affected). This procedure is very similar to the one of [105]. In a pure sampling phase (no updates of the parameters), the set of temperatures converges fairly quickly (about 100 MC steps), while successfully reaching flat swap rates histograms, see Fig. (4.8).

Adaptive APT

One major issue with RBM training is that the minimal number of Monte Carlo steps required to learn correctly the model varies a lot from one data-set to another, because some distributions are harder to thermalize than others. Interestingly, the same MoI used for sampling can be used for monitoring the mixing rate at almost no additional computational cost. First, given a Markov Chain $\mathbf{v}^{(t)}$, we assign to each sample its corresponding mode $z^{(t)} = \arg\max P_0(z|\mathbf{v})$. Then, we can monitor the transitions from one mode to another by measuring the transition matrix $Q(z, z') = \left\langle z^{(t)} = z | z^{(t-1)} = z' \right\rangle$. A well-mixing Markov Chain will exhibit large off-diagonal terms, for all z, whereas a bad mixing one will have mostly diagonal entries equal to one; this can be quantified by the effective convergence time $\tau = -\frac{1}{\log \lambda_2}$, where λ_2 is the second

largest eigenvalue of Q. Starting from $N_{MC} = 1$ step between each gradient update, we increase N_{MC} until $\tau < \tau_{max}$ where e.g. $\tau_{max} = 10$, and decrease N_{MC} by 1 if $\tau \leq \tau_{max} \frac{N_{MC}}{N_{MC}-1}$. Note that τ cannot be identified directly with the true convergence time of the Markov Chain, as $z^{(t)}$ is not itself a Markov Chain; τ is only a lower bound, as it does not take into account intra-mode thermalization time, nor memory effect such as back-and-forth swaps between replica. Nonetheless, it is easier to interpretate and much less costly to evaluate than the autocorrelation function.

4.2.3 *Results*

We first compare Gibbs, PT and APT sampling on a RBM with 400 dReLU hidden units trained on MNIST. Clearly, APT improves over PT: as seen from Fig. 4.9(a,b), samples at lower β are much closer to the target distribution, resulting in higher swap rates and shorter trajectories in the β ladder (Fig. 4.10). Quantitatively, the autocorrelation function Fig. 4.9(c) confirms that APT decorrelates samples faster than both Gibbs and PT, at about equal computational cost. Transition matrices displayed in Fig. 4.9(d) show that transitions between modes are significantly more frequent and homogeneously distributed with APT, resulting in smaller thermalization time τ.

In terms of training, as expected, APT can successfully learn the mixture model shown in Fig. (4.5) onto which both PCD and PT fail. Quantitative results on non-trivial data sets are discussed in the next chapter.

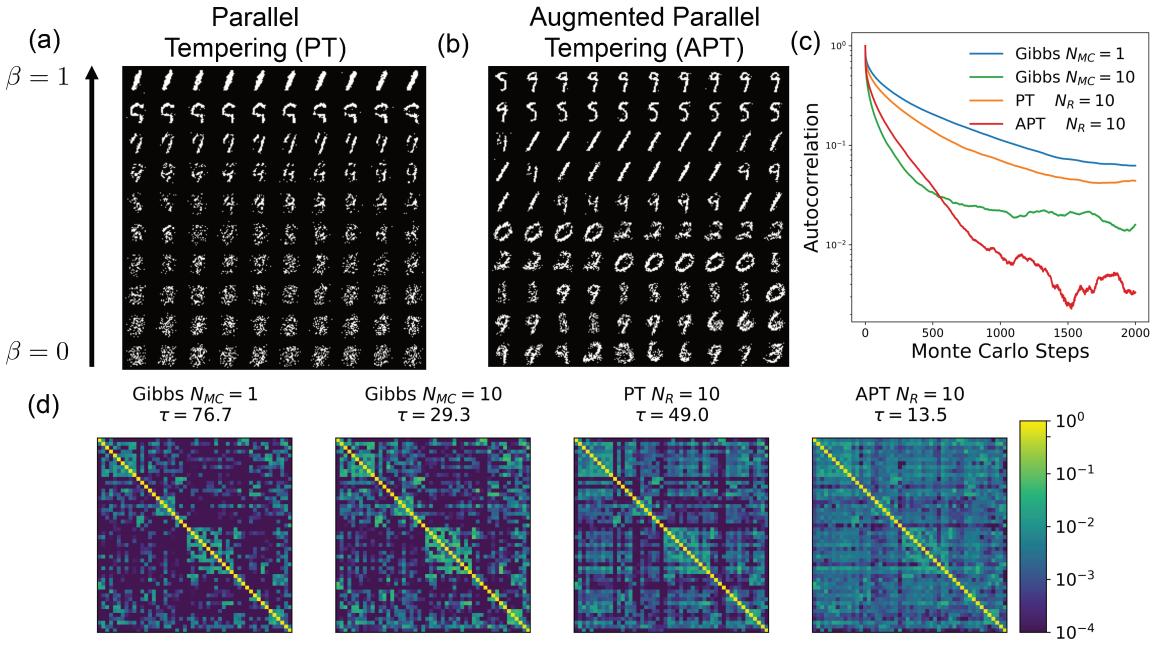

Figure 4.9: Samples produced by dReLU RBM $M = 400$ trained on MNIST, (a) using PT (b) using APT. (c) Markov Chains Autocorrelation Function $C(\tau) = \left(\sum_i P\left[v_i^{(t+\tau)} = v_i^{(t)}\right] - \left[P(v_i)^2 + (1 - P(v_i))^2\right]\right) / \left(2\sum_i P(v_i)(1 - P(v_i))\right)$ for Gibbs, PT and APT. (d) Transition matrices measured using the MoI shown in Fig. (2.2). Modes are ordered by similarity.

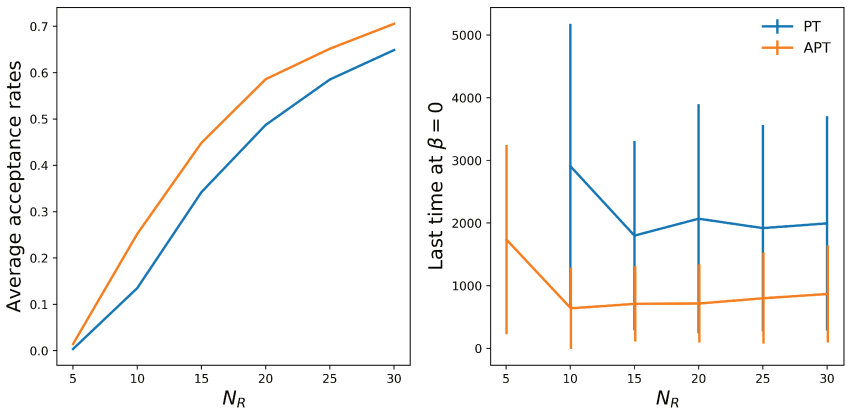

Figure 4.10: **Evaluating ergodicity in the β space for Parallel Tempering and Augmented Parallel Tempering**. Average acceptance rates (left) and last time since the $\beta = 1$ particle $\mathbf{v}_{\mathbf{N_R}}$ was at $\beta = 0$ (i.e. half of round-trip time). The minimum time (~ 700) is 3 times lower for APT than for PT (> 2000), and reached for a smaller number of replica N_R.

5

THE PARAMETERIZATION PROBLEM

5.1 BACKGROUND

Besides the gradient evaluation problem, maximum likelihood training of RBM is impaired by the non-convexity and bad conditioning of the hessian. We illustrate first the problem on few examples.

(a) *Performance drops with trivial data transformation*. As discussed in the literature [118], the naive SGD algorithm for RBM with Bernoulli hidden units gives much worse results on 1-MNIST (*i.e.* MNIST with all pixels flipped) than on MNIST, see likelihood curves in Fig. (5.1)A. As seen from panel B, this is because at the end of the training, a significantly larger number of hidden units are either always active or silent ($\langle h_\mu \rangle \sim 0/1$); these units are therefore essentially useless in practice, as they are compensated by the fields. This 'inactivation' can occur when the distribution of input I_μ shifts quickly (e.g. its mean increases) and the field g_μ are not compensated fast enough. Once a hidden unit is inactivated, both gradients over g_μ and $w_{i\mu}$ cancel out, such that it stays inactivated. For instance, a small random move $w_{i\mu} \to w_{i\mu} + 0.01 * N(0,1)$ yields a shift of mean of order $0.01 \times \sqrt{\sum_i \langle v_i \rangle^2}$. For MNIST and 1-MNIST, this gives respectively 6 and 24; moves are therefore much larger for the later and inactivation is more frequent. Reducing the learning rate could overcome this issue, but it would slow learning considerably.

(b) *Several ReLU hidden units do not learn* The same problem can be observed with ReLU hidden units. For a given sample, the initial input distribution $I_\mu \sim \mathcal{N}(0,1) \times 0.01 \times \sqrt{\sum_i \langle v_i \rangle^2}$, which can be fairly negative. If a hidden unit has initial inputs significantly below its threshold, its activity will be very small, and so are the gradients with respect to $w_{i\mu}$ and θ_μ. It will therefore learn slower than the others; if not at all. This is seen from the evolution of the weight amplitude $W_\mu = \sqrt{\sum_w i\mu^2}$ in Fig. (5.2): some weights grow very quickly whereas others lag behind. We note that this effect is dynamical rather than a characteristic of the optimum. Indeed, if

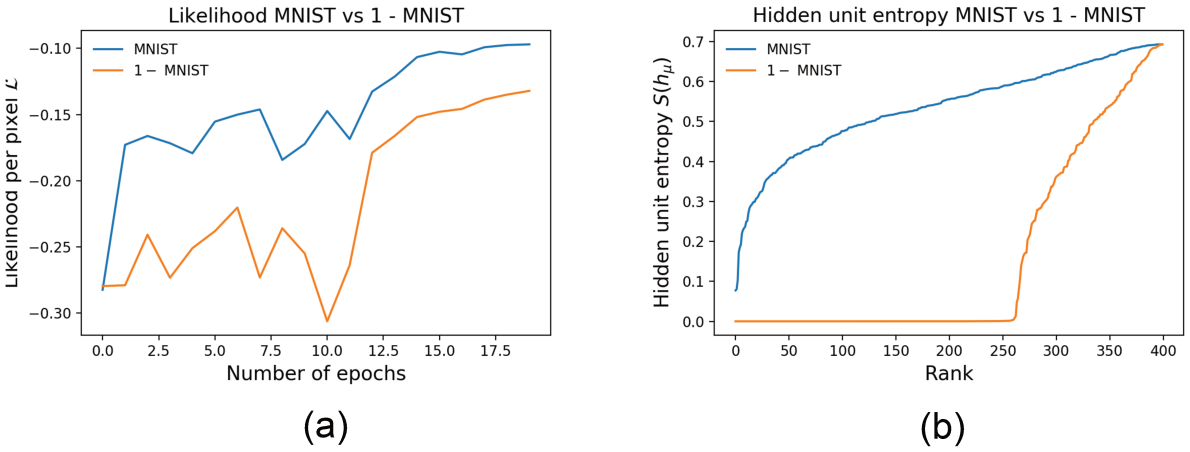

Figure 5.1: **Comparing Training on MNIST and 1-MNIST** A. Likelihood throughout training B. Distribution of hidden unit entropies $S(h_\mu) = -\langle h_\mu \rangle \log \langle h_\mu \rangle - (1 - \langle h_\mu \rangle) \log(1 - \langle h_\mu \rangle)$. Many hidden units are inactivated with 1-MNIST.

the wide distribution of weight amplitude was a property of the optimum (like in PCA where the distribution of variance drops quickly), the small components would be reproducible from one training to another, which is not the case.

(c) *RBM with continuous hidden units tend to diverge* Literature [119] and numerical experiments (see Table 5.1) report that RBM with continuous hidden units require much smaller learning rate in order to avoid divergence; which often makes them much less effective in practice. Consider for instance the case of Gaussian hidden unit. Since there is a redundancy between the parameters of the potential $\mathcal{U}_\mu(h) = \frac{\gamma_\mu h^2}{2} + \theta_\mu h$ and the weight amplitude and fields [1], we can set $\gamma_\mu = 1$, $\theta_\mu = 0$ $\forall \mu$ without loss of generality. This is the standard choice reported in the literature; although simple, it is such that the scale of the hidden units activity h_μ depends on the scale of the weights, w, and on the number of visible units, N. Indeed, $\Gamma'_\mu(I_\mu) \equiv \frac{I_\mu - \theta_\mu}{\gamma_\mu} = I_\mu$, which can get as large as $w\,N$ when a visible configuration is strongly overlapping with the weight vector. As the support of h is not bounded, neither are the gradients of the log-likelihood with respect to $w_{i\mu}(v)$, and divergence can occur. Moreover, the hessian is ill-conditioned as well: for instance, its diagonal elements $\partial^2_{w_{i\mu}} \sim h^2_\mu$ can range from 1 to $w^2 N^2$. Therefore, we also expect large discrepancy between its highest and lowest eigenvalues, such that small learning rates

1 the model distribution is invariant under rescaling transformations $\gamma_\mu \to \lambda^2 \gamma_\mu$, $w_{i\mu} \to \lambda w_{i\mu}$, $\theta_\mu \to \lambda \theta_\mu$ and offset transformation $\theta_\mu \to \theta_\mu + K_\mu$, $g_i \to g_i - \sum_\mu w_{i\mu} \frac{K_\mu}{\gamma_\mu}$

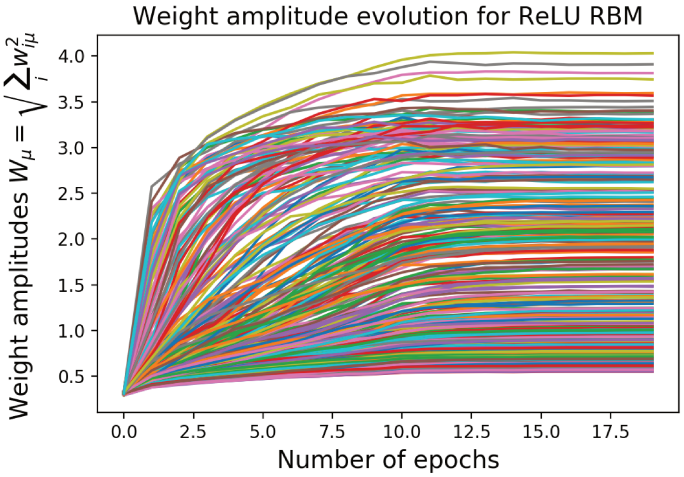

Figure 5.2: Evolution of weight amplitude for all hidden units in a ReLU RBM trained on MNIST. Some units learn very quickly and many others not at all.

are required to achieve convergence. Lastly, the support of h_μ fluctuates a lot during training (unlike Bernoulli hidden units), which complicates the moment estimation and can harm convergence.

In general, these defects, typical of first order methods, can be overcome by using quasi-second order methods such as L-BFGS [120] or adaptive learning rates such as momentum [121] and ADAM [20]. For instance, in ADAM optimization, the learning rate is adapted to the scale of each parameter component; allowing smaller updates for more sensitive parameters and vice-versa. Unfortunately, all these method require taking differences of gradients at successive time steps - either to estimate the hessian or to evaluate the variance of the update direction. This is not possible for RBM trained by PCD/PT/APT, as the successive gradients estimations are correlated owe to the model averages evaluated by MCMC. Hence taking difference of successive gradients or low-pass filtering it can give very inaccurate direction estimates.

Best improvements were obtained by methods that rely on reparametrization of the model, such as the enhanced gradient [118] or the centering trick [122] for Bernoulli hidden units; we propose to generalize this approach to continuous hidden units.

5.2 A NEW REPARAMETERIZATION TRICK FOR RESTRICTED BOLTZMANN MACHINES

In both Bernoulli and Gaussian hidden units, changes in one parameter must be accompanied by another change to maintain the hidden unit activity. This covariate shift phenomenon [123] is general to all neural network, and an interesting solution, batch normalization, has been recently proposed for feed-forward networks [21]. The idea is to reparametrize the network such that all intermediate activities have zero mean and unit variance. For the quadratic potential, we adapt this idea and choose γ_μ and θ_μ such that:

$$\left\langle h_\mu(\mathbf{v}) \right\rangle_d = 0 \,, \quad \mathrm{Var}[h_\mu(\mathbf{v})]_d = 1 \tag{5.1}$$

where Var denotes the variance. These implicit equations over γ_μ, θ_μ can be solved analytically:

$$\gamma_\mu = \frac{1 + \sqrt{1 + 4\,\mathrm{Var}[I_\mu(\mathbf{v})]_d}}{2} \,, \quad \theta_\mu = \left\langle I_\mu(\mathbf{v}) \right\rangle_d \equiv \sum_i w_{i\mu} \left\langle v_i \right\rangle_d \tag{5.2}$$

Since γ_μ, θ_μ must be updated after each SGD step and evaluating $\mathrm{Var}[I_\mu(\mathbf{v})]_d$ using the entire data set is computationally expensive, we compute it using only the current mini-batch (before performing the gradient update), and use an exponential moving average over γ_μ. Moreover, since γ_μ, θ_μ are functions of $\mathbf{w}$, the gradients with respect to the weights must be updated accordingly as:

$$\frac{\partial}{\partial w_{i\mu}} \mathcal{L} \leftarrow \frac{\partial}{\partial w_{i\mu}} \mathcal{L} + \frac{\partial \gamma_\mu}{\partial w_{i\mu}} \frac{\partial}{\partial \gamma_\mu} \mathcal{L} + \frac{\partial \theta_\mu}{\partial w_{i\mu}} \frac{\partial}{\partial \theta_\mu} \mathcal{L} \tag{5.3}$$

Which gives, after taking derivative of Eqn. 5.2:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_{i\mu}} &= \left\{ \left\langle v_i h_\mu \right\rangle_d - \left\langle v_i \right\rangle_d \left\langle h_\mu \right\rangle_d \right\} - \left\{ \left\langle v_i h_\mu \right\rangle_m - \left\langle v_i \right\rangle_d \left\langle h_\mu \right\rangle_m \right\} \\ &+ \frac{\mathrm{Cov}\left[I_\mu(\mathbf{v}), v_i \right]_d}{\sqrt{1 + 4\mathrm{Var}[I_\mu(\mathbf{v})]_d}} \left\{ \left\langle h_\mu^2 \right\rangle_d - \left\langle h_\mu^2 \right\rangle_m \right\} \end{aligned} \tag{5.4}$$

Eqn. 5.4 generalizes the centering trick to Gaussian hidden units. Crucially, the normalization condition Eqn. 5.1 guarantees that each gradient (and hessian)

component are of order 1 regardless of the size of the network, contrary to Eqn. 4.1. The same idea can be adapted for dReLU potentials, but with significant technical complication. First, the potential must be reparameterized to express the two invariance conditions in terms of two independent parameters. Second, the condition $\langle h \rangle_d = 0$ cannot be satisfied without loss and generality and must be slightly changed. Lastly, solving analytically $\mathrm{Var}[h]_d$ is impossible due to the non-linearity of the transfer function; instead a fixed-point equation over γ_μ is derived. Interested reader are refered to Annex A.3 for the technical details.

5.3 RESULTS

We now evaluate generative performance improvements. RBM with various potentials, sampling methods and SGD parameters were trained on the MNIST and Caltech Silhouettes dataset. We run SGD for 40 (resp. 1000) epochs, with an initial learning rate lr, and an a exponential decay of the learning rate to $\mathrm{lr}_f = 10^{-3}\mathrm{lr}$ after 25% of the updates. The partition function is evaluated using Annealed Importance Sampling [96], $n_\beta = 510^4, M = 100$. As shown from Tables 5.1,5.2, training benefits from dReLU hidden units, better sampling and reparametrization. The most important factor is the choice of hidden unit potentials and parametrization. Surprisingly, better sampling is not crucial for these two data sets provided that the learning rate is properly annealed. Compared to over studies, the values of up to -65 nats for MNIST are significantly higher than most reports in the literature which are around -80 nats, both with undirected graphical models (RBM, DBM,...) [124] and variational autoencoders [125, 126]. Similarly, 100 continuous hidden units performs better for Caltech Silhouettes than 4000 Bernoulli hidden units (-107 nats as reported in [118]). These results must be taken with caution, as the AIS procedure can result in large overestimation of the likelihood. However, we did find consistent results with i) larger M, n_β, ii) another implementation of AIS and iii) the reverse AIS procedure, which underestimates the likelihood. This suggests that despite their simple architecture, RBM can be very effective generative models provided a proper training algorithm is used.

Table 5.1: Test set likelihood on MNIST for several hidden unit potentials, learning rates and with (left) or our without reparametrization (right)

	$\mathrm{lr} = 10^{-3}$	$\mathrm{lr} = 10^{-2}$	$\mathrm{lr} = 5.10^{-2}$	$\mathrm{lr} = 10^{-1}$
Gaussian	-102.1/ -104.3	-81.2/-85.9	-76.8/ Diverge	-75.9/Diverge
dReLU	-96.6/-88.7	-71.2/-73.0	-71.7/ Diverge	-71.0 /Diverge

Table 5.2: Test set likelihood for MNIST (left) and 28×28 Caltech Silhouettes 101 dataset (right). Standard deviations of order 0.1 (resp. 0.5)

	Gibbs ($N_{MC} = 1$)	Gibbs ($N_{MC} = 10$)	PT ($N_{PT} = 10$)	APT ($N_{PT} = 10$)	Adaptive APT
Bernoulli	-70.7/-145.0	-70.4/-142.7	-70.5/-143.6	-70.4/-143.0	**-70.2**/**-142.4**
Gaussian	-77.0/-105.7	**-76.0**/ **-104.4**	-76.9/-105.8	-76.8/-105.7	-76.7/ -105.2
dReLU	-68.0/-106.2	**-65.7**/-106.2	-67.3/-105.9	-66.8/-106.1	-66.4/**-105.5**

DISCUSSION

Overall, training BM and RBM is not obvious: numerous methods exist, with their advantage and liabilities. If one were to start reimplementing RBM from scratch, our experiments suggest that:

- Careful initialization matters. We initialize fields from the independent model, and weights as $\mathcal{N}\left(0, \sigma^2 = \frac{0.01}{N}\right)$, so as to avoid early hidden units saturations.

- Parameterization is critical: highest performance and best results were obtained using a dynamic reparametrization trick, chosen such that hidden units have $\langle h_\mu \rangle \sim 0$, $\mathrm{Var}[h_\mu] = 1$. When using it, dReLU potentials systematically outperform Bernoulli and quadratic potentials. If no regularization over the weights is used, the simple Gaussian gauge is enough. Otherwise, a lengthy computation is unfortunately required, see Annex A.3.

- In general, Persistent Contrastive Divergence, combined with a proper annealing procedure of the learning rate is a fairly reliable gradient approximation scheme. One must however play with the number of gradient updates until a satisfactory result is reached, as there is currently no way to know in advance how many updates are required. Moreover, no simple early stopping procedure exists, as the likelihood is noisy and costly to evaluate. At the expense of tougher programming, gains - or at least guarantees - can be obtained using Parallel Tempering or Augmented Parallel Tempering, a new method introduced in this thesis. APT is a principled way to overcome the main limitation of MCMC sampling, namely its inability to explore efficiently multimodal, non-connected data distribution. APT consistently outperforms PT, but we have yet to find real-life situations where APT finds results that cannot be obtained by using PCD with many Gibbs steps, many gradient updates and slowly annealing learning rate.

Several directions can improve the current developments. First, in our APT implementation, the mixture model is fitted first to the data, and remains fixed. We have also tested variants where the mixture model is dynamically fitted to the RBM samples rather than to the data; swap rates were slightly higher but

similar results were obtained. An interesting application of this procedure is the case where $N >> M$: we can dynamically fit a MoI to the hidden layer, which could allow decent thermalization even when N is very large. Another possibility is to use Coupled Simulated Tempering [124] instead of Parallel Tempering; it could significantly reduce the computational burden.

Altogether, we have reported several algorithmic improvements for training RBM that significantly improve their generative performance. Interestingly, for the MNIST data set, the likelihood scores from RBM are significantly higher than in the scores reported in the literature that were obtained with deep directed graphical models such as variational autoencoders. Though a careful validation of the likelihood scores and further experiments are required, this work suggests that shallow undirected models, which are often easier to interpretate can perform on-par with deep directed graphical models. Future prospects include generalizing to other undirected graphical models, and more broadly designing new learning algorithm that associate the expressivity of undirected graphical models with the sampling efficiency of directed graphical models.

Part III

STATISTICAL MECHANICS OF RESTRICTED BOLTZMANN MACHINES

We have presented in Section 3.6 a rich phenomenology of behaviors for RBM trained on real data. When tested on MNIST, a non-trivial real data-set, RBM can learn different types of representations depending on the training and model parameters. Moreover, in some cases, they can learn surprisingly well such complex data-distribution despite their very simple architecture. This phenomenology raises several questions. Firstly, how can such simple networks generate a complex distribution with a large variety of local minima, matching the original data points? Secondly, why do some hidden unit potentials give good results, whereas others do not? Lastly, can we connect this behavior to the one of the Hopfield model, corresponding to the case of quadratic hidden unit potential? It is hopeless to provide analytical answers to these questions in full generality for a given RBM with parameters fitted from real data. However, statistical physics methods and concepts allow us to study the typical energy landscape and properties of RBM drawn from appropriate random ensembles. In this part, we will first review some background on network-based models of associative memory. We then present an ensemble of Random-RBM inspired from Hopfield's work and based on the main properties of RBM trained on real data. We present theoretical results and phase diagram, and compare theoretical predictions with RBM trained on real data.

6

BACKGROUND ON NETWORK-BASED ASSOCIATIVE MEMORY MODELS

6.1 THE HOPFIELD MODEL OF ASSOCIATIVE MEMORY

We present first the Hopfield network, originally studied by Little in 1974 [127] and popularized by Hopfield in 1982 [65]. The original task was to design content-addressable memory systems based on brain-like parallel architectures - and conversely, understand how neural network could store memories. It was initially defined as the following dynamical system. We consider a set of N binary (McCulloch–Pitts) neurons, with associated activities $S_i \in \{-1, 1\}$ [1], modeled as either silent $S_i = -1$ or spiking at maximum rate $S_i = 1$. Let $J_{ij}, 1 \leq i < j \leq N$ a neural connectivity matrix modeling the synaptic couplings. Positive and negative entries correspond respectively to excitatory and inhibitory synapses. We consider the following asynchronous evolution of neural activity:

$$S_i \leftarrow \text{Sign}(\sum_{j \neq i} J_{ij} S_j) \equiv \text{Sign}(I_i) \tag{6.1}$$

Eqn. (6.1) defines a dynamical system in which each neural state is updated depending on its input received: the neuron is activated when the total input is positive and not otherwise. Let $\xi_{i\mu} \in \{-1, 1\}, \mu = 1..M$ a finite set of M 'patterns' of neural activity that ought to be stored within the system. For simplicity, we assume here that each memory entry is drawn randomly and independently from the others $P(\xi_{i\mu} = 1) = P(\xi_{i\mu} = -1) = \frac{1}{2}$. Then provided that:

$$J_{ij} = \sum_{\mu} \xi_{i\mu} \xi_{j\mu} \tag{6.2}$$

1 We use -1/1 notations for consistency with the following AGS computation

Each memory state $S_i = \xi_{i\mu}$ is a fixed point of the dynamical system. Indeed, starting from a memory state with e.g. $\mu = 1$, the input received by neuron i writes:

$$\begin{aligned} I_i &= \sum_{j \neq i} J_{ij}\xi_{j1} = \sum_{j \neq i, \mu} \xi_{j1}\xi_{j\mu}\xi_{i\mu} \\ &= \xi_{i1}(N-1) + \sum_{j \neq i, \mu \neq 1} \xi_{j1}\xi_{j\mu}\xi_{i\mu} \end{aligned} \tag{6.3}$$

Since each memory is drawn randomly from the others, they are quasi-orthogonal from one another, such that $\sum_l \xi_{i\mu}\xi_{i\mu'} \sim \sqrt{N}\mathcal{N}(0,1)$. It follows that the first term in Eqn. (6.3) is of order N and same sign as ξ_{i1}, whereas the second term is of order $\sqrt{NM} << N$ if M is finite (or of order $\log N$) and N is large. Therefore, $\text{Sign}(I_i) = \xi_{i1}$ and the pattern is stable. Interestingly, the memory is also marginally stable: starting from a neural state sufficiently close to ξ_{i1} (essentially with overlap $\sum S_i\xi_{i1} \sim \mathcal{O}(N)$), the network dynamic also converges to the memory state. The Hopfield network can therefore be seen as a content-addressable memory: a dynamical system in which each memory is stored as an attractor that can be queried with an initial cue (a small part of memory) to retrieve memories.

This new memory concept raises several questions. Firstly, how robust is the memory with respect to noise, as is the case in biological neural networks. Indeed, biological neurons are inherently stochastic, and the deterministic update Eqn. (6.1) is not realistic. A more reasonable model is to assume a probabilistic response, in which the neuron is activated with logistic probability:

$$P(S_i = 1|I_i) = \frac{1}{1 + e^{-\beta I_i}} \tag{6.4}$$

The noise level is controlled by the 'inverse temperature' β: small and large β correspond respectively to pure noise and pure deterministic behaviors. Secondly, what is the capacity of the memory system, i.e. how many systems can be retrieved? As seen from Eqn. (6.3), trouble is expected when M is of order N. Thirdly, is the system specific, i.e. are all attractors of the dynamical system stored patterns? In particular, it is easy to show that the so-called spurious states, defined as:

$$\Xi_{i,\{\mu_1,\mu_2,\mu_3\}} = \text{Sign}\left(\xi_{i\mu_1} + \xi_{i\mu_1} + \xi_{i\mu_1}\right) \tag{6.5}$$

are stable under the noiseless dynamics, despite a finite overlap with all three patterns ($\sum_i \Xi_{i,\{\mu_1,\mu_2,\mu_3\}}\xi_{i\mu_1} \sim 0.5N$). More generally, any finite combination of L patterns with L odd is also a stable state of the dynamic. This suggests that false memories could arise, but the subsequent error rate in memory retrieval is unknown. In his original paper, Hopfield addressed these questions with numerical simulations, and showed in particular that (i) the memory is stable with respect to noise and asymmetry in neuronal couplings J_{ij} (ii) the system can store about $M \sim 0.15N$ patterns (iii) most initial configurations converge toward one of the original patterns.

6.2 STATISTICAL MECHANICS OF ASSOCIATIVE MEMORY NETWORKS

The first theoretical results supporting these observation were obtained by Amit, Gutfreund and Sompolinsky (AGS) in 1985 [26,128,129] using the replica theory, a statistical mechanics tool developed for studying disordered materials [130]. The connection with statistical mechanics between the Hopfield model can be seen as follows. We first define the following Hamiltonian over neural configurations $\mathbf{S} \in \{-1,1\}^N$ and its associated Boltzmann distribution:

$$\begin{aligned} H(\mathbf{S}) &= -\frac{1}{2N}\sum_{i,j}\left(\sum_{\mu}\xi_{i\mu}\xi_{j\mu}\right)S_iS_j \\ P_T(\mathbf{S}) &= \frac{e^{-\frac{1}{T}H(\mathbf{S})}}{Z_T} \end{aligned} \tag{6.6}$$

Within this framework, the noiseless and noisy update rule of Eqn. (6.1,6.3) directly correspond to the zero-temperature and the finite temperature (with $\beta = \frac{1}{T}$) Monte Carlo Gibbs update of the probability distribution (6.6). In particular, a recall trajectory using the noisy update converges toward a sample from the probability distribution (6.6). Therefore, the questions raised above can be answered by studying what are the 'typical' macroscopic configuration states dominating the system, and how do they vary with T and $\alpha \equiv \frac{M}{N}$. In this language, the Hopfield network behaves correctly as an associative memory if the probability distribution concentrates around the original patterns ξ_{μ}. This way, a dynamic with any initial configuration will quickly converge around one of the patterns.

In particular, the StatMech framework allows to understand easily why the spurious states are not problematic in practice for finite M. First, we rewrite the Hamiltonian as:

$$H = -\frac{N}{2}\sum_{\mu} m_{\mu}\left(\mathbf{S}\right)^2 \tag{6.7}$$

Where the overlap (or magnetization) is defined as $m_{\mu}\left(\mathbf{S}\right) = \frac{\sum S_i \xi_{i\mu}}{N} \in [-1, 1]$. Their energy can be computed analytically as:

$$E_L = -\frac{L}{2^{2L-1}}\begin{pmatrix} L-1 \\ \frac{L-1}{2} \end{pmatrix}^2 \tag{6.8}$$

Which is significantly higher for $L > 1$ (e.g. $E_3 = -\frac{3N}{4}$) than for the memory states $L = 1$ ($E_1 = -\frac{1}{2}$). Intuitively, this comes from the 'winner-takes-all' structure of the Hamiltonian. Since the patterns are essentially orthogonal and the neural state has fixed norm N, one cannot have, say, both $m_1 = 1$ and $m_2 = 1$. In fact, starting from ξ_1 and switching one at a time the S_i from ξ_{i_1} to ξ_{i_2} produces a set of configurations with $|m_1| + |m_2| \sim 1$. By convexity of $x \to x^2$, the lowest energies are reached on the border $m_1 = 1, m_2 = 0$ or $m_1 = 0, m_2 = 1$, i.e. on the patterns. Therefore, the spurious states are almost non-existent provided that N and the inverse temperature β are sufficiently large. On the other hand, the case of $M = \mathcal{O}(N)$ is trickier, because though each spurious state has higher energy than the original patterns, their number is exponentially increasing with N, and they could be entropically favored. AGS addressed this question by assuming random uniform patterns and computing the average (over the patterns) free energy landscape of the model. They deduced the following phase diagram, reproduced in Fig. (6.1):

- A paramagnetic phase, dominant at high temperature, in which the system is mostly driven by noise. The system essentially explores almost uniformly the set of configurations, and the vast majority of these configurations have weak overlap with all patterns.
- A ferromagnetic or retrieval phase, dominant at low temperature and low ratio α, in which the system focuses on configurations near the patterns. Starting from any random configuration, the system quickly converges toward one of the patterns.
- A Spin Glass phase, similar to the one of the Sherrington-Kirkpatrick model [64], in which a large number of metastable states with weak overlap with all patterns dominate.

- A metastable retrieval state, in which both ferromagnetic phase and spin glass phase coexist, but the latter is dominant. Memory retrieval is still possible provided the initial overlap with the memory is large enough.

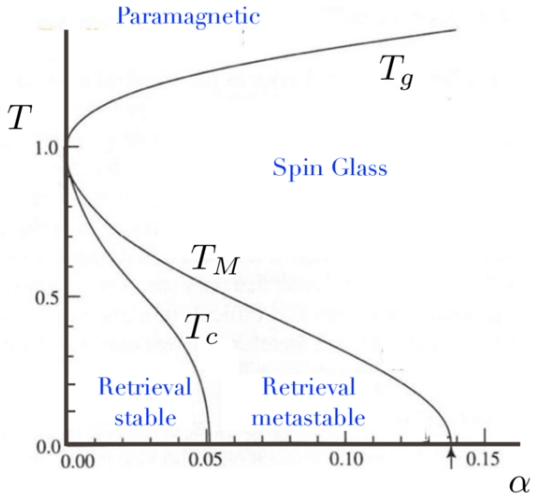

Figure 6.1: Phase diagram of the Hopfield model. Derivation presented in [26], and diagram reproduced from [131]

In particular, AGS find at $T = 0$ a critical capacity $\alpha_c = 0.138$, in very good agreement with the original numerical experiments of Hopfield. The AGS computation sparked interest in the study of associative memory models. Several researchers investigated more biologically plausible models [132], how non-symmetrical or correlated patterns could be stored [133–135], and how other learning approaches could increase the total capacity of the network. A notable example is the pseudo-inverse rule, another way to encode memories into synaptic coupling [136, 137] showing robustness to correlations and increased capacity. Irrespective of the learning rule, Gardner showed that the maximum capacity was $\alpha_c = 2$ [25], a number that can be reached by the perceptron learning algorithm.

6.3 MULTITASKING IN ASSOCIATE MEMORY NETWORKS

In the following two decades, most of the theoretical neuroscience community moved on to study other network models, such as feedforward architectures or chaotic neural networks, which are not relevant for our work. We move away from neuroscience and jump in time to a recent series of papers by Agliari, Barra

and collaborators introduced in the context of immunology [138,139,139,140]. In this series of works, a connection was suggested between i) the Hopfield model and ii) Networks of interacting T-cells and B-cells; in this context, memory retrieval corresponds to pathogen recognition and response. Contrary to the original Hopfield model, several pathogens must be retrieved (and dealt with) simultaneously. Agliari et al. showed that provided that the patterns (*i.e.* interaction matrix between the set of T-cells and B-cells) are sparse, taking values $+1, -1$ with probability $\frac{p}{2}$ or 0 with small probability p, the network is able to retrieve several patterns/pathogen simultaneously. The key factor is that sparsity breaks the 'winner-takes-all' structure of the Hamiltonian discussed previously. Since patterns are weakly overlapping, one can find configurations with both high m_1, m_2, m_3, see Fig. 6.2. This allows multi-memory states such as the spurious states described in Eqn. (6.1) to become thermodynamically favorable against single memory states.

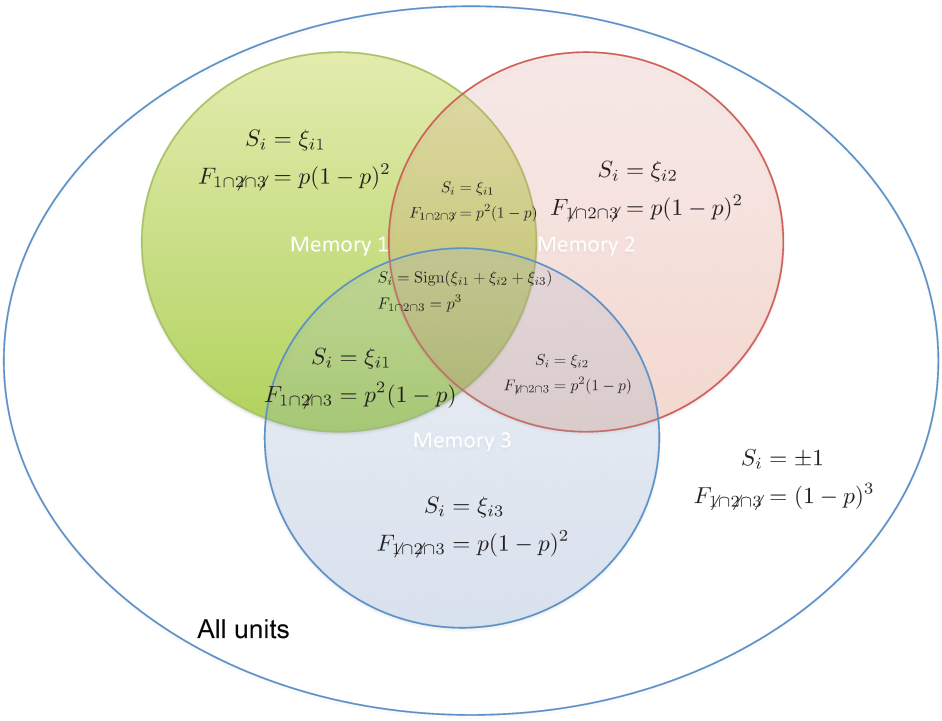

Figure 6.2: **Example of multi-patterns configuration** Each memory $\xi_{i\mu}$ defines a region R_μ of units with $|\xi_{i\mu}| \neq 0$ (colored circle). A configuration taking $S_i = \xi_{i\mu}$ in the regions where there is no conflict and a spurious state-like $S_i = \sum'_\mu \xi_{i\mu'}$ in overlapping regions is constructed. It has lower energy than a single memory $S_i = \xi_{i1}$ if $i \in R_\mu$, ± 1 otherwise.

Agliari et al. proposed different Ansatz for the ground state of the system, such as the Parallel States and Hybrid States which have lower energy than the spurious states, and studied phase transitions. We show the set of rescaled

magnetizations $\frac{m_1}{p}, .., \frac{m_L}{p}$ found as function of p for various Ansatz in Fig. 6.3. The main take-away message is that as p decreases (*i.e.* the sparsity increases), they reach lower energy than the single-memory state. Moreover, they match pretty closely as $p \to 0$, suggesting a universal behavior at low p.

The main limitation of the proposed model is that it cannot be generalized to an infinite number of patterns $M \sim N$. Indeed, there is no mechanism preventing the explosion of the number of patterns in practice. For instance, in a model with finite connectivity $p = \frac{c}{N}$, Sollich et al. showed a critical capacity $\alpha_c \sim c^{-2}$ [140]. For $\alpha < \alpha_c$, the connectivity is so diluted that the graph is not connex - such that multitasking occurs but is trivial, whereas for $\alpha \geq \alpha_c$, a giant component emerges and the system falls in a spin glass phase. Our Random-RBM ensemble, which we will introduce can overcome this issue.

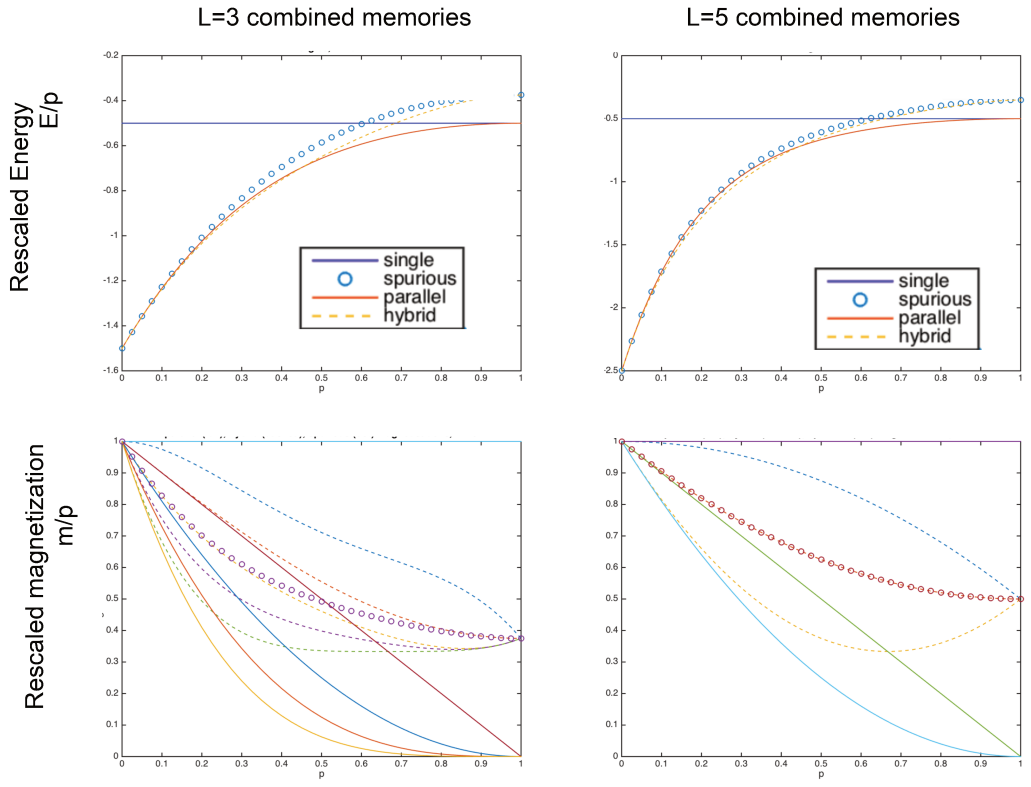

Figure 6.3: **Energy and magnetization of combined states** Rescaled energy $\frac{E}{p}$ (top) and magnetizations $\frac{m}{p}$ of the single memory configuration and various multiple memory configurations involving $L = 3, 5$ patterns. Each line style (full, dot, dash), denotes a different configuration Ansatz. For the magnetization, magnetizations m_μ are not equal for all patterns in the multiple state and each color is one of the L magnetization values.

7

THE RANDOM-RBM MODEL

7.1 MODEL DEFINITION

7.1.1 *Main model ingredients*

In Part 3.6, we presented results of training on the MNIST handwritten data set. We review here the main observations:

(a) *Importance of non-linearity* Non-linear models perform better than linear ones, and learn meaningful representation.

(b) *Sparse weight matrix* Shortly after the beginning of the training, weights are similar to digits; as training converges, each weight focus on individual strokes. The weight matrix $w_{i\mu}$ becomes sparser, and with larger entries, see Fig. 3.9. At the end of training, we find a fraction of non-zero weights $p \sim 0.1$, see Section 8.2 for details of the estimation.

(c) *Low effective temperature* As training converges, the system is effectively at very low temperature. This can be seen from the conditional average $\langle v_i | \mathbf{h} \rangle$, shown in Fig. 7.1A: all conditional averages are essentially either black (0) or white (1), with very few grey (intermediate) values. Another way to see this is to evaluate the pseudo-likelihood:

$$\mathcal{PL} = \left\langle \log P(v_i | \{v_j, j \neq i\} \right\rangle \tag{7.1}$$

The pseudo-likelihood quantifies how much a component varies given the other components. As seen from Fig. 7.1B it is fairly close to zero, suggesting that for a given configuration, all the visible units are essentially frozen.

(d) *Compositional Representations* At the end of the training, the hidden layer representation shows a compositional behavior, with $1 \ll L \ll M$ strongly activated hidden units for each sample. At the end of training, see Section 8.2 for details of the estimation.

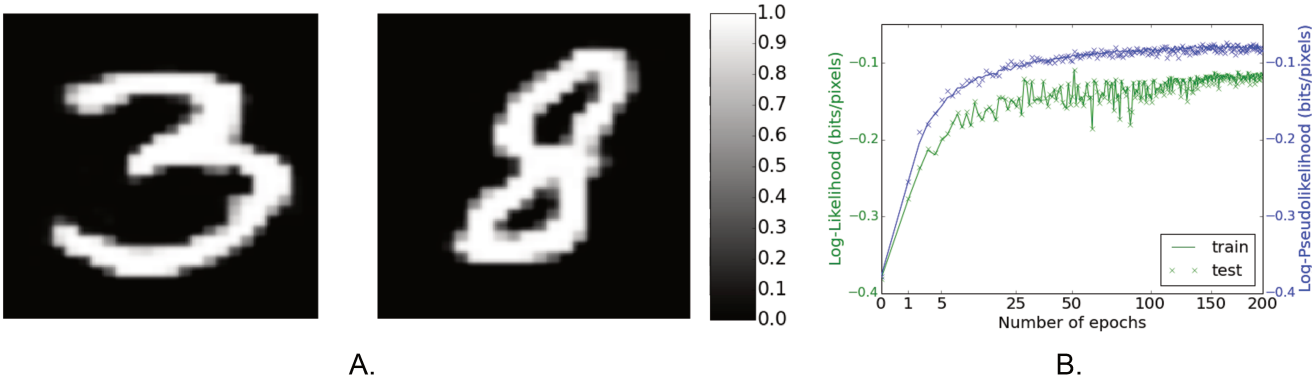

Figure 7.1: A. Conditional means $\langle v_i|\mathbf{h}\rangle$ for two hidden units configurations sampled at equilibrium. Most pixels v_i are frozen, with $\langle v_i|\mathbf{h}\rangle \, in\{0,1\}$ B. Evolution of the per-pixel log-likelihood and pseudo-likelihood of the model. After training, each unit value is almost completely determined by the value of the other units.

(e) *Complex Energy landscape* The number of attractors steadily increases from 1 to a very large number, $\gg M$.

A last observation which will be important for quantitative fit is that the degree of connectivity of the visible units is heterogeneous. As seen from Fig. 3.8 some pixels, e.g. are the border are almost disconnected from the hidden layer whereas others are connected to a large number of hidden units.

7.1.2 *Random-RBM ensemble*

Inspired by these observations, we define the following Random-RBM ensemble model for ReLU hidden units:

- N binary visible units, M ReLU hidden units. We will take the thermodynamic limit $N, M \to \infty$ where $\alpha = \frac{M}{N}$ is finite.
- uniform visible layer fields, i.e. $\mathcal{U}_v(v_i) = -gv_i, \ \forall i.$
- uniform hidden layer thresholds, $\mathcal{U}_h = \begin{cases} \frac{h^2}{2} + \theta h & \text{If } h \geq 0 \\ +\infty & \text{Otherwise} \end{cases}, \ \forall \mu.$
- a random weight matrix $w_{i\mu} = \frac{\xi_{i\mu}}{\sqrt{N}}$, where each 'pattern' $\xi_{i\mu}$ is drawn independently, taking values $+1, -1$ with probabilities $\frac{p_i}{2}$ and 0 with

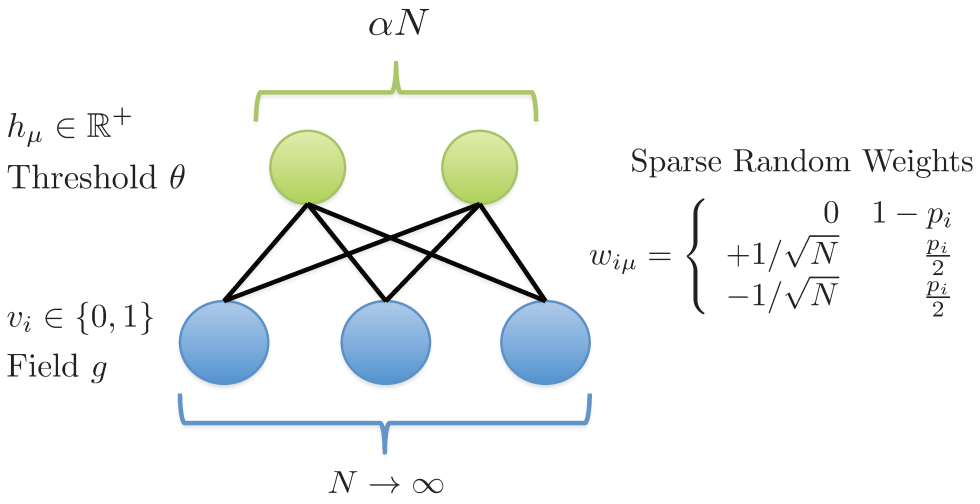

Figure 7.2: The Random-RBM ensemble, with its control parameters: threshold θ of hidden ReLU, ratio α of the sizes of the hidden and visible layers, field g on visible units, sparsities p_i of the weights.

probability $1 - p_i$. The *degree of sparsity* $p = \frac{\sum_i p_i}{N}$ is the fraction of non-zero weights.

Given a realization of the weight matrix, the energy and probability writes (for $h_\mu > 0 \; \forall \mu$):

$$\begin{aligned} E(v,h) &= -g \sum_{i=1}^{N} v_i + \frac{1}{2} \sum_{\mu=1}^{\alpha N} h_\mu^2 + \theta \sum_{\mu=1}^{\alpha N} h_\mu - \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \sum_{\mu=1}^{\alpha N} \xi_{i\mu} v_i h_\mu \\ P_\beta(v,h) &= \frac{e^{-\beta E(v,h)}}{Z_\beta} \end{aligned} \tag{7.2}$$

The control parameters of our model are α, $p \equiv \frac{\sum p_i}{N}$, g and θ. We recall that several variants or special cases have already been addressed in the literature. Choosing Gaussian hidden units, full patterns ($p_i = 1 \; \forall i$) and ± 1 visible units with $g = 0$ leads exactly to the original Hopfield model, see the equivalence between Gaussian RBM and the Hopfield model in Eqn.(3.6). Choosing Gaussian hidden units, uniform sparse weight distribution $p_i = p < 1 \; \forall i$ and ± 1 visible units with $g = 0$ corresponds to the model of Agliari et al.

7.1.3 *The Hopfield model revisited*

We briefly reinterpret the behavior of the Hopfield model in light of this connection. In the Gaussian-RBM model, each hidden unit is associated with a single pattern. Indeed, from Eqns. (7.2) and (3.4) we identify pattern magnetizations (overlaps) and hidden unit inputs:

$$I_\mu = \sqrt{N} m_\mu \tag{7.3}$$

A recall state $\mathbf{v} = \xi_{i1}$ therefore corresponds to a hidden layer with hidden layer activity $h_1^*(\mathbf{v}) = \sqrt{N}$ (see definition in Eqn. (3.7, 3.8)). For all remaining hidden units, $m_\mu = \frac{1}{N}\sum_i \xi_{i1}\xi_{i\mu} \sim \sqrt{N}\mathcal{N}(0,1)$, such that $h_\mu^*(\mathbf{v}) \sim \mathcal{N}(0,1)$. To evaluate the stability of the recall state, let us write the outcome of a single Gibbs sampling update at zero temperature:

$$\begin{aligned} \mathbf{v} \to \mathbf{h} &= \arg\max P(\mathbf{h}|\mathbf{v}) = (\sqrt{N}, h_2^*, .., h_M^*) \\ \mathbf{h} \to \mathbf{v} &= \arg\max P(\mathbf{v}|\mathbf{h}) = \Theta\left(\xi_{i1} + \frac{1}{\sqrt{N}}\sum_{\mu\neq 1}\xi_{i\mu}h_\mu^*\right) \end{aligned} \tag{7.4}$$

The recall state will be stable provided that the total input I_i received by each visible unit is dominated by the one received from hidden unit 1. This is true if $\frac{1}{\sqrt{N}}\sum_{\mu\neq 1}\xi_{i\mu}h_\mu^* \sim \frac{M}{N} < 1$, e.g. for finite M. Clearly, choosing a non-linear transfer function such as ReLU is a also very simple way to suppress this undesired input. A ReLU with a threshold θ of order 1 can silence off most of the $h_\mu^*, \mu \neq 1$, (e.g. $\theta = 2.58$ to suppress about 99% of the hidden units), while not changing h_1 at leading order.

We note that one should be cautious and not attempt to derive an estimate of the critical capacity from this single equation only: as $N, M \to \infty$, there is always at least one component v_i whose input is dominated by the noise rather than by h_1, such that the initial state is not perfectly stable. This may increase the value of the others h_μ, which could in turn flip additional units, and so on, until the pattern is unstable. To obtain a rigorous metastability limit, we must study the free energy landscape with a replica computation, as in the Hopfield model.

Similarly, this back-and-forth sampling also allows to better understand the process of recalling a pattern from a partial cue. Starting from a visible layer

configuration with small overlaps m_μ, we compute the hidden layer activity $\mathbf{h}$. Then, the visible units receive a larger input from the strongest hidden unit $h_{\max}$, and will tend to align to the corresponding pattern ξ_μ. The overlap $m_{\max}$ increases, and at the next iteration the dominant hidden unit will be even more strongly activated. The hidden units therefore follow a 'winner-take-all' dynamic, where hidden units compete for magnetization, and the one with largest initial overlap tends to reach maximum value. In this regard, the spurious states with L strong magnetizations correspond to an equilibrium between L competing hidden units; and it is clearly unstable. We can also interpret the Spin Glass phase from this perspective: when M is too large, no single hidden unit can dominate over the others, such that all magnetizations m_μ are weak.

To illustrate how hidden units effectively interact with one another, we can compute the effective energy $E_{\text{eff}}(\mathbf{h}) \equiv -\log P(\mathbf{h})$ by summing over the visible layer configurations $\mathbf{v}$, as was done for computing $P(\mathbf{v})$ in Eqn. 3.3. We get:

$$E_{\text{eff}}(\mathbf{h}) = \frac{1}{2}\sum_\mu h_\mu^2 - \sum_i \log\cosh\left(\sum_\mu w_{i\mu} h_\mu\right) \tag{7.5}$$

Then, a 4th order Taylor expansion of the log cosh gives:

$$E_{\text{eff}}(\mathbf{h}) = \frac{1}{2}\sum_\mu h_\mu^2 - \frac{1}{2}\sum_{\mu,\mu'}\left(\sum_i w_{i\mu} w_{i\mu'}\right) h_\mu h_{\mu'} + \frac{1}{2}\sum_{\mu,\mu'} h_\mu^2 h_{\mu'}^2 \left(\sum_i w_{i\mu}^2 w_{i\mu'}^2\right) + ... \tag{7.6}$$

Where we have not written the terms in h_μ^3, h_μ^4. This expansion sheds light on two points. First, strong positive overlaps between patterns induce strong pairwise positive couplings; it is therefore difficult to find recall state configurations with one strongly activated hidden unit and not the other overlapping ones. This explains the observation by Hopfield that strongly correlated patterns tend to merge in practice, and cannot be recovered individually. Second, the binary nature of the visible units (which is responsible for high-order terms) induces an effective repulsion term between each pair of hidden units. The magnitude of the repulsion depends on the overlap between the supports of the hidden units: the repulsion is maximal for the Hopfield model ($\sum_i w_{i\mu}^2 w_{i\mu'}^2 = \frac{1}{N}$) and much smaller when the patterns are sparse ($\sum_i w_{i\mu}^2 w_{i\mu'}^2 = \frac{p^2}{N}$). This illustrates why hidden unit may compete or collaborate depending on the statistics of the weight matrix.

7.2 REPLICA COMPUTATION AND MEAN-FIELD EQUATIONS

We now sketch the main steps of the replica computation. The goal it to compute the average free energy of the model:

$$f(\alpha, \{p_i\}, g, \theta) \equiv \lim_{N\to\infty} -\frac{1}{\beta N} \overline{\log Z\left(\alpha, \beta, \{p_i\}, g, \theta, \{\xi_{i\mu}\}\right)}\,, \tag{7.7}$$

where the overline denotes the average over the $\{\xi_{i\mu}\}$ and the partition function reads

$$Z\left(\alpha, \beta, \{p_i\}, g, \theta, \{\xi_{i\mu}\}\right) = \sum_{\mathbf{v}\in\{0,1\}^N} \int \prod_{\mu=1}^{M} dh_\mu \; e^{-\beta E(\mathbf{v},\mathbf{h})}\,. \tag{7.8}$$

In particular, our goal is to evaluate the free energy as function of the number L of strongly activated hidden units, *i.e.* with $h_\mu = \sqrt{N}m$ as in the Hopfield model, and of their associated magnetizations (supposed identical for simplicity). Following the replica trick $\overline{\log Z} = \lim_{n\to 0} \frac{\overline{Z^n}-1}{n}$, we write the partition function for n replica (indexed by a) of the system sharing the same quenched weights:

$$Z^n = \sum_{v_i^a} \int \prod_{\mu,a} dh_\mu^a e^{\sum_{\mu,a} \beta \mathcal{U}_h(h_\mu^a) + \beta \sum_{i,a} \mathcal{U}_v(v_i^a) - \beta \sum_{i,\mu} w_{i\mu} \sum_a v_i^a h_\mu^a} \tag{7.9}$$

We fix $h_1^a = h_2^a = ... = h_L^a = m\sqrt{N}$, and assume the others are of order 1. As in the AGS computation, we treat both subsets separately, and average over the quenched disorder on the second subset first:

$$\overline{\exp\left(\beta \sum_{i,\mu\geq L+1,a} v_i^a h_\mu^a w_{i\mu}\right)} \sim \prod_{i,\mu\geq L+1} \exp\left(\frac{\beta^2 p_i}{2N} \sum_{a,b} v_i^a v_i^b h_\mu^a h_\mu^b\right) \tag{7.10}$$

Where we have used $\mathrm{Var}(w_{i\mu}) = \frac{p_i}{N}$, and have kept the extensive term in N only. We formally decouple the visible and hidden layer by introducing the order parameters:

$$q^{ab} = \frac{\sum_i p_i v_i^a v_i^b}{\sum_i p_i} \equiv \frac{1}{N} \sum_i \frac{p_i}{p} v_i^a v_i^b \tag{7.11}$$

And its Lagrange conjugate: $\hat{q}^{ab}$, so as to replace the integrand as follows:

$$\begin{aligned}
&\sum_{v_i^a} \phi(\sum_i \frac{p_i}{p} v_i^a v_i^b, , \mathbf{h^a}, \mathbf{h^b}, ...) \\
&= \int \prod_{a \leq b} \frac{d\hat{q}^{ab} dq^{ab}}{2\pi/N\beta} \phi(q^{ab}, \mathbf{h^a}, \mathbf{h^b}, ...) \sum_{v_i^a} \exp\left[-\beta N \sum_{a \leq b} \hat{q}^{ab} \left(q^{ab} - \sum_i \frac{p_i}{Np} v_i^a v_i^b \right) \right]
\end{aligned} \tag{7.12}$$

After some manipulation, the partition function rewrites:

$$\begin{aligned}
\overline{Z^n} = \int \prod_{a \leq b} \frac{d\hat{q}^{ab} dq^{ab}}{2\pi/N\beta} \exp\left[-\beta N \sum_{a \leq b} \hat{q}^{ab} q^{ab} - \beta n L \times \mathcal{U}_h(m\sqrt{N}) \right] \\
\times \prod_i \left(\sum_{v_i^a} \exp\left[-\beta \sum_a \mathcal{U}_v(v_i^a) + \beta \sum_{a \leq b} \hat{q}^{ab} v_i^a v_i^b \frac{p_i}{p} + \beta m \left(\sum_{\mu=1}^{L} \sqrt{N} w_{i\mu} \right) \left(\sum_a v_i^a \right) \right] \right) \\
\times \left(\prod dh_a \exp\left[-\beta \sum_a \mathcal{U}_h(h^a) + \frac{\beta^2 p}{2} \sum_{a,b} q^{ab} h^a h^b \right] \right)^{\alpha N - L}
\end{aligned} \tag{7.13}$$

We now assume a replica-symmetric Ansatz, with both q_{ab},$\hat{q}_{ab}$ taking only two values (on diagonal and off-diagonal). They are parameterized as follows:

$$\begin{aligned}
q_{ab} &= q + \delta_{a,b} \frac{C}{p\beta} \\
\hat{q}_{ab} &= \frac{\alpha\beta p}{2} \left[2r(1 - \delta_{a,b}) + \delta_{a,b} \left(r + \frac{B}{p\beta}) \right) \right]
\end{aligned} \tag{7.14}$$

Where q, r, B, C are assumed to have finite limit as $\beta \to \infty$. Note that in $\hat{q}_{ab}$, we have $2r$ in the off-diagonal because the matrix has only lower diagonal indices $a \leq b$. The parameterization is justified by their interpretation, see Section 7.2.1.

Then the three lines of Eqn. (7.13) can be computed. The first term gives, at leading order in β, N and n:

$$1 - n\beta L \times \mathcal{U}_h\left(m\sqrt{N}\right) - \frac{n\beta N\alpha}{2} \left[qC + rB \right] + \mathcal{O}(n^2) \tag{7.15}$$

The third term writes:

$$\begin{aligned}
&\prod_a dh_a \exp\left[-\beta\sum_a \mathcal{U}_h(h^a) + \frac{\beta^2 p}{2} q(\sum_a h_a)^2 + \frac{\beta C}{2}(\sum_a h_a^2)\right] \\
&= \int Dz \left(\int dh \exp\left[-\beta\mathcal{U}_h(h) + \beta\sqrt{qp}zh + \frac{\beta C}{2}h^2\right]\right)^n \\
&= 1 + n\int Dz \log\left(\int dh \exp\left[-\beta\mathcal{U}(h) + \beta\sqrt{qp}zh + \frac{\beta}{2}Ch^2\right]\right) + \mathcal{O}(n^2) \\
&\approx 1 - n\beta\int Dz \min_h\left[\mathcal{U}(h) - \sqrt{qp}zh - \frac{C}{2}h^2\right] + \mathcal{O}(n^2)
\end{aligned} \tag{7.16}$$

Where Dz denote a Gaussian measure $\frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}$, and where the last line was obtained by a saddle point approximation of the integral over h.

Similarly, the second one gives:

$$1 - n\beta\int Dz \min_v\left[\mathcal{U}_v(v) - \left[m\left(\sum_{\mu=1}^{L}\sqrt{N}w_{i\mu}\right) + z\sqrt{\alpha p_i r}\right]v - \frac{\alpha}{2}B\frac{p_i}{p}v^2\right] + \mathcal{O}(n^2) \tag{7.17}$$

Finally, summing all contributions and keeping the first order term in n,N,β gives the following free energy value:

$$\begin{aligned}
f(\alpha, \{p_i\}, g, \theta) = &\frac{L}{2}m^2 + \frac{\alpha}{2}(q\,B + r\,C) \\
&- \frac{1}{N}\sum_i \left\langle \int Dz \min_v\left[\mathcal{U}_v(v) - (mW + z\sqrt{\alpha p_i r})v - \frac{\alpha}{2}B\frac{p_i}{p}v^2\right]\right\rangle_{W=\sqrt{N}\sum_{\mu=1}^{L}\xi_{i\mu}} \\
&\alpha\int Dz\ \min_h\left(\mathcal{U}_h(h) - \frac{C}{2}h^2 - z\sqrt{pq}\,h\right)
\end{aligned}$$

Where we used $\mathcal{U}_\mu(x) \sim_{x\to\infty} \frac{x^2}{2}$, valid for both quadratic and ReLU potentials. The weight sum $W = \sqrt{N}\sum_{\mu=1}^{L} w_{i\mu}$ takes integer values between $-L$ and L.

7.2.1 *Interpretation of the order parameters*

Here, we have expressed the free energy as function of the number L and magnetization m of the strongly activated hidden units, and have introduced

another set set of order parameters in the process: q, r, C, B. What is their physical interpretation? By taking the derivative of Eqn. (7.13) with respect to $q_{ab}, \hat{q}_{ab}$, we find a simple relation between the saddle point values of the order parameters and the moments of the Random-RBM distribution:

$$\begin{aligned} q &= \overline{\frac{1}{N}\sum_i \frac{p_i}{p}\langle v_i\rangle^2} \approx_{\beta\to\infty} \overline{\frac{1}{N}\sum_i \frac{p_i}{p}\langle v_i\rangle} \\ r &= \overline{\frac{1}{M-L}\sum_{\mu>L}\langle h_\mu\rangle^2} \\ C &= \lim_{\beta\to\infty} \overline{\frac{\beta p}{N}\sum_i \frac{p_i}{p}\langle v_i\rangle\left(1-\langle v_i\rangle\right)} \\ B &= \lim_{\beta\to\infty} \overline{\frac{\beta p}{M-L}\sum_{\mu>L}\left\langle h_\mu^2\right\rangle - \langle h_\mu\rangle^2} \end{aligned} \tag{7.18}$$

In other words, q is the (weighted) mean activity in the visible layer, r is the square average activation of the weakly activated hidden units, and C,B are the rescaled variance of the visible (resp. hidden) units. C and B can also be interpreted as susceptibilities. Indeed, we have the following fluctuation-dissipation relationship (similar as in Eqn. (3.8)):

$$\frac{\partial < h_\mu | I_\mu >}{\partial I_\mu} = \frac{\partial}{\partial I_\mu}\frac{\int dh^\mu e^{-\mathcal{U}_\mu(h_\mu)} e^{\beta h^\mu I_\mu} h_\mu}{\int dh^\mu e^{-\mathcal{U}_\mu(h_\mu)} e^{\beta h^\mu I_\mu}} = \beta \mathrm{Var}(h_\mu | I_\mu) \tag{7.19}$$

And similarly for the visible layer. Therefore, we have:

$$\begin{aligned} C &= \overline{\frac{p}{N}\sum_i \left\langle \frac{\partial v_i}{\partial I_i}\right\rangle} \\ B &= \overline{\frac{p}{M-L}\sum_\mu \left\langle \frac{\partial h_\mu}{\partial I_\mu}\right\rangle} \end{aligned} \tag{7.20}$$

C, B measure local gains, i.e. how a small additional input of one layer to the other affect the activity of the other layer.

7.2.2 *Saddle-point equations*

The previous computation was general for any hidden unit potential; we now focus on ReLU. We further assume that the p_i have a density of the form $\rho(\frac{p_i}{p})$, such that the average over visible sites can be replaced by an integral over $x = \frac{p_i}{p}$. Let:

$$H^{(k)}(x) = \int_{z=x}^{\infty} Dz(z-x)^k \tag{7.21}$$

The free energy rewrites:

$$\begin{aligned} f =& \frac{L}{2}m^2 + \frac{\alpha}{2}(q\,B + r\,C) \\ & -\sqrt{\alpha p r}\int \rho(x)\sqrt{x}\left\langle H^{(1)}\left(-\left[g + \frac{\alpha}{2}B\,x + m\,W\right]/\sqrt{\alpha\,p\,x\,r}\right)\right\rangle_W dx \\ & -\frac{\alpha p q}{2(1-C)}H^{(2)}\left(\frac{\theta}{\sqrt{pq}}\right) \end{aligned} \tag{7.22}$$

Deriving with respect to m,r,q,B,C gives the following set of self-consistency equations for the order parameters:

$$m = \frac{1}{L}\int \rho(x)\left\langle W H^{(0)}\left(-\left[g + \frac{\alpha}{2}B\,x + m\,W\right]/\sqrt{\alpha\,p\,x\,r}\right)\right\rangle_W dx \tag{7.23}$$

$$q = \int \rho(x)\left\langle H^{(0)}\left(-\left[g + \frac{\alpha}{2}B\,x + m\,W\right]/\sqrt{\alpha\,p\,x\,r}\right)\right\rangle_W dx \tag{7.24}$$

$$C = \frac{\sqrt{p}}{\sqrt{2\pi\alpha r}}\int \rho(x)\left\langle \exp\left(-\frac{1}{2}\left[g + \frac{\alpha}{2}B\,x + m\,W\right]^2/\alpha\,p\,x\,r\right)\right\rangle_W dx \tag{7.25}$$

$$r = pq/(1-C)^2 H^{(2)}(\theta/\sqrt{pq}) \tag{7.26}$$

$$B = \frac{p}{1-C}H^{(0)}(\theta/\sqrt{pq}) \tag{7.27}$$

Briefly, we explain how to solve numerically these equations. In the case where $p_i = p\ \forall i$, we use the change of variables $M = \frac{m}{\sqrt{\alpha p r}}$, $\theta_v = -\frac{g+\frac{\alpha B}{2}}{\sqrt{\alpha p r}}$, $\theta_h = \frac{\theta}{\sqrt{pq}}$. The above equations can be rewritten so as to express all order parameters m,r, q,B,C and model parameters α,g,θ as function of M,θ_v,θ_h only. Therefore,

instead of varying α, g, θ and solving the equations by fixed-point iteration, we directly scan through M,θ_v,θ_h to obtain a set of model/order parameter pairs of values $(\alpha, g, \theta), (m, q, r, B, C)$. In the general case where the degree distribution is not uniform, we make the approximation $g + \frac{\alpha}{2} B\, x + m\, W \sim g + \frac{\alpha}{2} B + mW$, justified when B is small (e.g. for large threshold), and repeat the procedure.

7.3 RESULTS

7.3.1 *Effect of the non-linearity*

We first study the non-sparse case with $p = 1$ and a single pattern $L = 1$, such that $W = \pm 1$. Moreover, we set the fields as $g = -\frac{\alpha B}{2}$ to compensate for the asymmetry of 0-1 units and the ReLU. [1]. By symmetry $H^{(0)}(x) + H^{(0)}(-x) = 1$, we recover $q = \frac{1}{2}$, and the equations over B is decoupled from the other, such that we find three equations over m,C and r very similar to the original AGS computation (see Eqn 6.41-6.43 page 300 of [66]):

$$\begin{aligned} m &= \frac{1}{2}\text{erf}\left(\frac{m}{\sqrt{2\alpha r}}\right) \\ C &= \frac{1}{\sqrt{2\pi r}} \exp\left[-\frac{m^2}{2\alpha r}\right] \\ r &= \frac{1}{2(1-C)^2} H^{(2)}(\theta\sqrt{2}) \end{aligned} \tag{7.28}$$

The only difference lies in the factors $\frac{1}{2}$ in the magnetization (because $m \in [-\frac{1}{2}, \frac{1}{2}]$ for 0-1 units) and $\frac{1}{2}H^{(2)}(\theta/\sqrt{2})$ in the noise equation. The later very quickly decays as $\theta \to \infty$, and illustrates how the thresholds damps the activity of the weakly activated hidden units. Since the change of variables $m \to \tilde{m} = \frac{m}{2}$, $r \to \tilde{r} = \frac{2r}{H^{(2)}(\theta/\sqrt{2})}$ and $\alpha \to \tilde{\alpha} = \frac{\alpha}{2H^{(2)}(\theta\sqrt{2})}$ gives back the exact equations as in AGS, the phenomenology is qualitatively identical with the original Hopfield model, featuring a ferromagnetic phase and a spin glass phase. The critical capacity is given by:

$$\alpha_c(\theta) = \frac{\alpha_c^{AGS}}{2H^{(2)}(\theta\sqrt{2})} \tag{7.29}$$

1 If we have +1/-1 spins and a symmetric ReLU such as the dReLU1 graph in Fig. 3.6, then setting $g = 0$ would give the same results

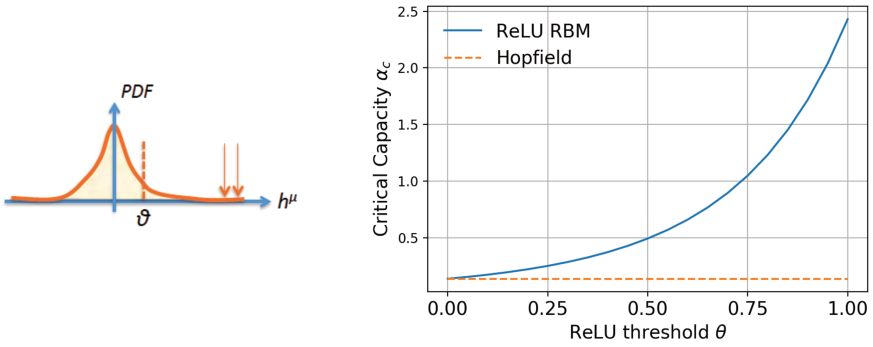

Figure 7.3: Critical capacity of ReLU RBM as function of the threshold. Thresholding cuts off the activity of the weakly activated hidden units (left), resulting in higher storate capacity

The capacity strongly increases with the threshold: as the level of crosstalk due to nonmagnetized hidden units diminishes larger ratios α can be supported by R-RBM without entering the glassy phase. For instance, $\theta = 0$ gives $\alpha_c = \alpha_c^{AGS} = 0.138$, and $\theta = 0.5$ gives $\alpha_c = 0.5$, see Fig. 7.3. Such values of α are closer to the typical values used in practice for RBM. As θ increases, we reach very quickly infinite capacity $\alpha_c = \infty$. It is however not necessarily a good idea in practice for a model to let $\theta \to \infty$ at fixed α, as it reduces the size of the basins of attraction of the model: the larger θ, the larger the initial overlap required to ensure convergence to the pattern.

7.3.2 $p < 1$ *and the compositional phase*

For small p, in addition to the ferromagnetic phase ($L = 1$) and the spin glass phase ($m = 0$), a new intermediate qualitative behavior emerges: a compositional phase, in which visible configurations have strong overlap with L features, where $1 \ll L \ll M$, see Fig. 7.4. To see this, we solve numerically the equations for fixed $L > 1$, using $p = 0.1$ $g = -0.02$ and varying θ. We show results in Fig. 7.5. For each value of L, there is a limit value $\alpha_c(L)$ above which a solution with $m > 0$ cannot exist. $\alpha_c(L)$ decays quickly with L but as in the non-sparse case, it can be made arbitrarily large by adjusting the threshold of the ReLU function. Interestingly, the normalized magnetization $\tilde{m} = m/(p/2)) \in [0, 1]$ decays relatively slowly with L, unlike in the Hopfield model; the same effect is observed in the parallel and hybrid states described by Agliari et al. at low p. Therefore, unlike the Hopfield model, solutions with $L > 1$ have lower free energy than the $L = 1$ phase (ferromagnetic) or spin glass phase ($m = 0$); see section below. All these observations are in very

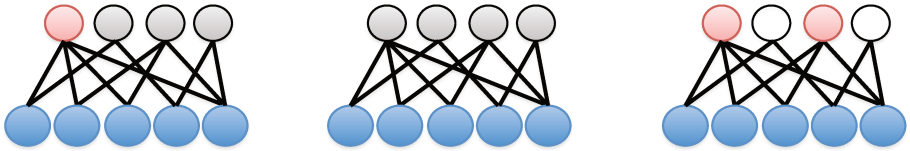

Figure 7.4: The three regimes of operation of Random RBM, see text. Red, grey and white hidden units symbolize, respectively, strong ($h \sim \sqrt{N}$), weak ($h \sim \pm 1$) and null ($h = 0$) activations. In the Ferromagnetic phase, one hidden unit is strongly activated and the others are weakly activated. The number of attractors is linear in N. In the spin glass phase, all hidden units are weakly activated, and there is a large number of metastable states. In the compositional phase, several hidden units are strongly activated, and the others are quiet. The number of attractors is polynomial in N

good agreement with numerical simulations of Random-RBM instances with $N = 10^4$ [2]. Starting from an initial configuration with a small finite overlap with L patterns, we quickly converge toward a configuration in which the L patterns are strongly activated and not the others. The magnetizations are not exactly identical for each pattern, but the average magnetization $\frac{\sum_{\mu=1}^{L} m_\mu}{L}$ matches well the theoretical prediction m. Moreover, as observed for RBM trained on real data, a combinatorial diversity of local minima is obtained by different choices of $\{\mu_1, ..., \mu_L\}$ of strongly activated hidden units.

7.4 RANDOM-RBM IN THE HIGH SPARSITY LIMIT $p \to 0$

7.4.1 *Scaling law and limit regime*

The nature of the large–L phases and the selection of the value of L are best understood in the limit case of highly sparse connections, $p \ll 1$. The R-RBM model exhibits an interesting limit behaviour, which we call hereafter compositional phase. In this regime the number of strongly magnetized hidden units is unbounded, and diverges as $L = \ell/p$, with $\ell > 0$ and finite. The strongly magnetized hidden units have a magnetization $m = \frac{p}{2}\tilde{m}$ with $\tilde{m}$ finite; *i.e.* such that the normalized between the visible layer configuration and the

2 Such large number is required to limit the deviation from finite-size effects, which are fairly important in practice. Simulations were performed on a GPU programmed with Theano [141]

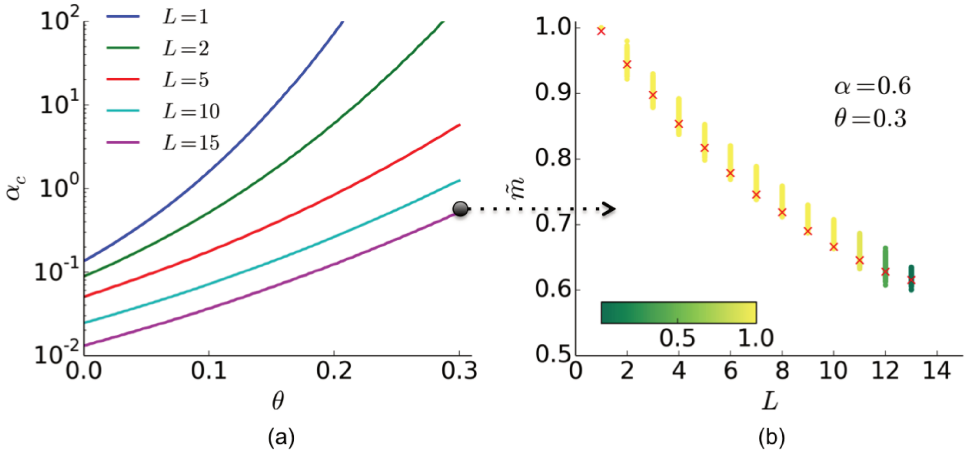

Figure 7.5: **Compositional regime in R-RBM. (a)** Critical lines in the θ, α plane below which L hidden units may be strongly activated. Parameters are $p_i = p = 0.1$, $g = -0.02$. **(b)** Comparison of theoretical (red crosses) and numerical simulations ($N = 10^4$, colored points) for the rescaled magnetizations $\tilde{m} = m/(p/2))$ as a function of the number L of strongly activated hidden units in R-RBM. 7,500 zero temperature MCMC, each initialized with a visible configuration strongly overlapping with $L = 1, 2, ...$ 'features', were launched; color code indicates the probability that the same L hidden units are magnetized after convergence (see bottom scale), and the corresponding average magnetization $\tilde{m}$.

weight is $\frac{\sum_i w_{i\mu} v_i}{\sum_i |w_{i\mu}|} \sim 1$. [3]. Nonmagnetized hidden units have activities of the order of $\sqrt{r} \sim \sqrt{p}$, and can be shutdown by choosing thresholds $\theta \sim \sqrt{p}$; hence crosstalk between those units can be suppressed, allowing for large relative size α of the hidden layer. The input received by a visible unit from the large number of magnetized units is, after transmission through the dilute weights, of the order of $L\, m\, p = \frac{1}{2}\, \ell^*\, \tilde{m}\, p$; it can be modulated by a (positive or negative) field $g \sim p$ to produce any finite activity q in the visible layer, as soon as the effective temperature gets below $\sim p$.

$$\left\{ \begin{array}{rclrcl} \beta &=& \frac{\tilde{\beta}}{p}, & g &=& p\tilde{g} \\ \theta &=& \sqrt{p}\tilde{\theta}, & f &=& p\tilde{f} \\ m &=& \frac{p}{2}\tilde{m}, & L &=& \frac{\ell}{p} \\ r &=& p\tilde{r}, & B &=& p\tilde{B} \end{array} \right. \tag{7.30}$$

Under these scaling laws, the random weight variable $W_i = \sqrt{N} \sum_{\mu=1}^{\frac{l}{p}} w_{i\mu}$ is a sum over $\frac{l}{p}$ terms, with a fraction $p_i = x_i p$ are non-zero; in the limit $p \to 0$, its probability law is well defined, given by the modified Bessel function of the first kind:

$$\begin{aligned} P_{\ell x_i}(W_i = w \in \mathbb{Z}) &= e^{-l x_i} I_w(l x_i) \\ I_\alpha(x) &= \sum_{k=0}^{\infty} \frac{1}{k! \Gamma(k+\alpha+1)} (\frac{x}{2})^{2k+\alpha} \end{aligned} \tag{7.31}$$

Then, rescaled free energy and saddle-point equations have a well-defined limit:

3 Solutions with nonhomogeneous magnetizations m_μ, varying from one strongly activated hidden unit to another, give additional contributions to f of the order of p^2 with respect to the homogeneous solution $m_\mu = m$, and do not affect the value of e_ℓ .

$$m = \frac{2}{\ell}\int \rho(x)\left\langle W H^{(0)}\left(-\left[\tilde{g}+\frac{\alpha}{2}\tilde{B}\,x+\frac{1}{2}\tilde{m}\,W\right]/\sqrt{\alpha\,x\,\tilde{r}}\right)\right\rangle_{W\sim P_{\ell x}(W)} dx \quad (7.32)$$

$$q = \int \rho(x)\left\langle H^{(0)}\left(-\left[\tilde{g}+\frac{\alpha}{2}\tilde{B}\,x+\frac{1}{2}\tilde{m}\,W\right]/\sqrt{\alpha\,x\,\tilde{r}}\right)\right\rangle_{W\sim P_{\ell x}(W)} dx \quad (7.33)$$

$$C = \frac{1}{\sqrt{2\pi\alpha\tilde{r}}}\int \rho(x)\left\langle \exp\left(-\frac{1}{2}\left[\tilde{g}+\frac{\alpha}{2}\tilde{B}\,x+\frac{1}{2}\tilde{m}\,W\right]^2/\alpha\,x\,\tilde{r}\right)\right\rangle_{W\sim P_{\ell x}(W)} dx \quad (7.34)$$

$$\tilde{r} = q/(1-C)^2 H^{(2)}(\tilde{\theta}/\sqrt{q}) \quad (7.35)$$

$$\tilde{B} = \frac{1}{1-C}H^{(0)}(\tilde{\theta}/\sqrt{q}) \quad (7.36)$$

$$\tilde{f}_\ell = -\frac{1}{8}\ell\tilde{m}^2 - \tilde{g}q - \frac{\alpha}{2}(q\tilde{B}+\tilde{r}C) + \frac{\alpha q}{2}\frac{\tilde{\theta}}{\sqrt{q}}H^{(2)}\left(\frac{\tilde{\theta}}{\sqrt{q}}\right) \quad (7.37)$$

Where in the last line, we injected the saddle-point values of the order parameters in the rescaled free energy. Under this low p regime, the phases mentioned above correspond to the three possible scenarios:

- *S*pin glass phase The global minimum of f is reached with $m = 0$.

- *F*erromagnetic phase The global minimum is reached with $m > 0$, and $\ell^* = 0$. f_ℓ is an increasing function of l. Typical configurations have a single hidden unit activated.

- *C*ompositional phase The global minimum is reached with $m > 0$, and f_ℓ is a non-monotonous function of l, reaching its minimum at ℓ^*. Typical configurations have about ℓ^*/p simultaneously activated hidden units.

A phase diagram can be derived as function of $\alpha, \tilde{\theta}, \tilde{g}$. Briefly speaking, given α, $\tilde{\theta}$ should be large enough (as in the finite p case) and $|\tilde{g}|$ should be neither too large to penalize the ferromagnetic phase, nor too small to avoid the spin glass regime, see Fig. 7.6.

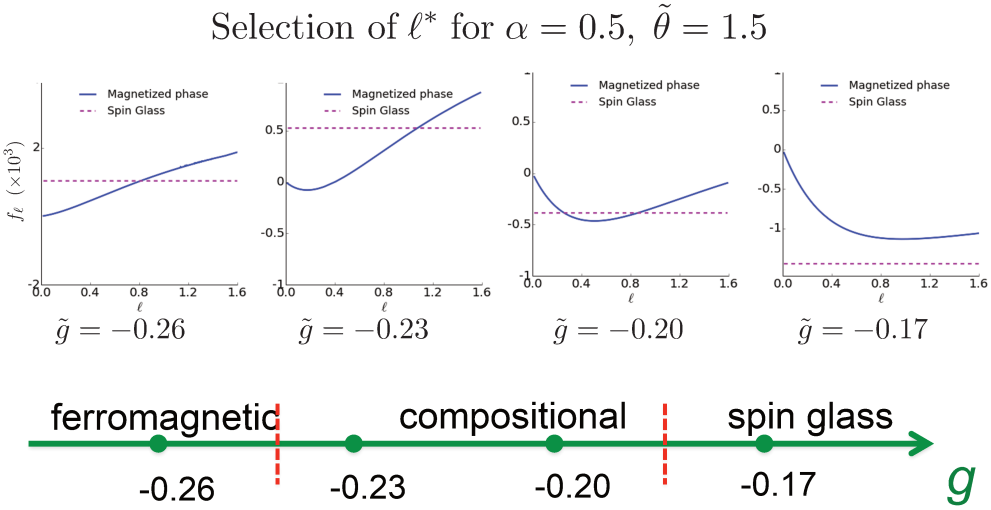

Figure 7.6: **Phase transitions in the low-p limit** For several values of $\tilde{g}$, curves f_ℓ, after optimizing over all the other order parameters (with $\tilde{m} > 0$, and comparison with the spin-glass free energy (with $\tilde{m} = 0$). As $\tilde{g}$ increases, two phase transitions occur.

7.4.2 *Geometry of the attractors*

We have therefore shown mathematically the existence of a compositional phase, which is characterized by a large diversity of local maxima $\mathbf{v}_k^*$ of $P(\mathbf{v})$ indexed by a subset $k = (k_1, ..., k_L)$ of $L^* = \ell^*/p$ strongly activated hidden units. We also found a traditional spin glass phase, in which there are also many metastable states, but hidden units are weakly activated. The two phases clearly differ by the coordinates in hidden layer representation of the local maxima $\mathbf{v}_k^*$, but are the marginal probability landscapes intrinsically different ? Indeed, two models may differ by hidden representations but not by their energy landscape. Consider for example the two following ensembles of Random-RBM:

- The traditional Hopfield model, with ± 1 visible units, $g = 0$, $p = 1$,$w_{i\mu} = \frac{\xi_{i\mu}}{\sqrt{N}}$ with random $\xi_{i\mu} = \pm 1$, and $\alpha < \alpha_c$.
- The *Rotated* Hopfield model, with ± 1 visible units, $g = 0$, $p = 1$, $w_{i\mu} = \frac{1}{\sqrt{N}} \sum_{\mu,\mu'} O_{\mu\mu'} \xi_{i\mu'}$, with a random orthonormal matrix O, and with random $\xi_{i\mu} = \pm 1$

By invariance of $W^T W$, the marginal $P(\mathbf{v})$ is identical, such that both models produce exactly the same set of local energy minima. In the first case however, the energy minima have a hidden layer representation with a single strongly

activated hidden units, whereas for the other case, due to the random rotation, they correspond to hidden units with weak (yet stereotyped) activities $O_{\mu 1}$ for all hidden units. Therefore, different representations do not necessarily mean different landscapes, and it is important to ask what differentiates the compositional and spin glass phase from the perspective of the visible layer only. Intuitively, the difference lies in the geometry of the local energy minima, see Fig. 7.7. In the ferromagnetic phase, the attractors take the form $\mathbf{v}_k^* \sim (1 + \xi_k)/2$, and are therefore all equidistant from one another, with Hamming distance $d(\mathbf{v}_k^*, \mathbf{v}_{k'}^*) = \frac{N}{2}(1 - \delta_{k,k'})$. In the spin glass phase, Parisi and collaborators showed that the metastable states were arranged in an ultrametric fashion, forming a hierarchy of clusters; some states are very close whereas other are very far away [142, 143]. On the other hand, we expect a different behavior in the compositional phase. Different attractors sharing a subset $\delta\ell/p$ of similar strongly activated hidden units should be gradually farther away from one another. This defines a different geometry in which distances are heterogeneous, but any pair of distant attractors $(\mathbf{v}_k^*, \mathbf{v}_{k'}^*)$ can be joined by a path of intermediate attractors sharing hidden units with both k,k', see Fig. 7.7.

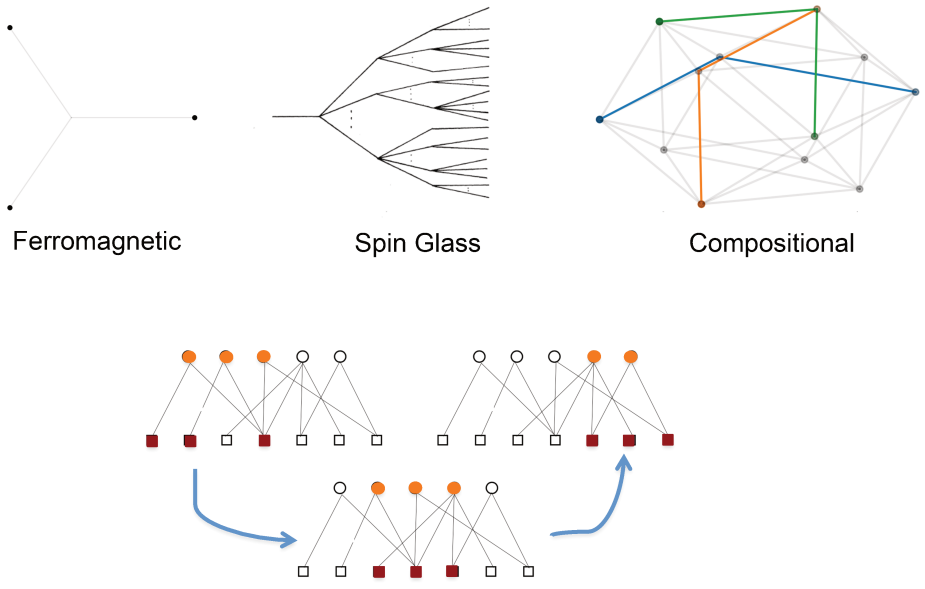

Figure 7.7: **Geometry of attractors** Top left: Ferromagnetic phase; all attractors are equidistant and far away. Top middle: Spin glass phase, reproduced from [143]; metastable states group into a hierarchy of clusters. Top right: Compositional phase; attractors are gradually away from one another, and can be linked by hopping from one attractor to the next. Bottom: Example of path between two distant attractors in the compositional phase

To validate this intuition, we propose to compute, through a real-replica approach, the average Hamming distance d (per pixel) between the visible configurations $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}$ minimizing the free energy (7.22) for two hidden configurations $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}$ sharing $(\ell - \delta\ell)/p$ hidden units among the ℓ/p strongly activated ones. We repeat the replica computation of the previous section but consider $2n$ replica of the system. In the first n replica, hidden units $\mu \in [1, L = \ell/p]$ are strongly activated, and in the second n replica, hidden units $\mu \in [L - L' = \frac{\ell - \delta\ell}{p}, 2L - L']$ are activated. The replica overlap $q_{ab} = \sum_i \frac{p_i}{Np} < v_i^a v_i^b >$ now takes three distinct values: $q + \frac{C}{\beta p}$ for $a = b$, q for $a \neq b$, with both $a, b \in [1, n]$ or $a, b \in [n+1, 2n]$, or q' if $a \in [1, n]$ and $b \in [n+1, 2n]$; q' denotes the average overlap between configurations sharing $\delta\ell/p$ strongly activated hidden units. Similarly, a hidden layer overlap between weakly activated hidden units $r' = \overline{< h_\mu^a >< h_\mu^b >}$ is introduced in the computation. Moreover, we add coupling terms of the form $J_v \sum_i \frac{p_i}{Np} v_i^a v_i^b$, and $J_h \frac{1}{M-L} \sum_{\mu > L} h_\mu^a h_\mu^b$ (with $a \leq n$ and $b > n$ or inversely) to the Hamiltonian. We then repeat the free energy computation and derivate with respect to J_v, J_h, in the limit $J_v, J_h \to 0$ to obtain equations over q', r'. Let $\eta = \frac{r'}{r}$, and $\phi = \frac{q'}{q}$. We obtain the following equations:

$$\begin{aligned}
\phi &= \int \rho(x) x dx \int Dz \left\langle \left\langle H^{(0)} \left[-\frac{\tilde{g} + \alpha \tilde{B} x/2 + 2\tilde{m}(W + W_1)}{\sqrt{\alpha \tilde{r} x (1-\eta)}} - z \sqrt{\frac{\eta}{1-\eta}} \right] \right\rangle^2_{W_1 \sim P_{\delta\ell x}(W_1)} \right\rangle_{W \sim} \\
\eta &= (1-\phi) \left(\int Dz H^{(2)} \left[\frac{\tilde{\theta}/\sqrt{q} - z\sqrt{\phi}}{\sqrt{1-\phi}} \right]^2 \right) \Big/ H^{(2)} \left[\tilde{\theta}/\sqrt{q} \right]
\end{aligned}$$

Numerical results are shown in Fig. 8.3. The Hamming distance D monotonously increases from $D = 0$ for $\delta\ell = 0$ up to $D = 2q(1-q)$ (complete decorrelation of visible units) for $\delta\ell = \ell$, in very good quantitative agreement with results for RBM trained on MNIST. Such gradual change have deep dynamical consequences. As seen from MCMC simulations of MNIST-trained RBM [4], gradual changes may occasionally lead to another digit type, by passing through well-drawn, yet ambiguous digits. The progressive replacement of feature-encoding hidden units (small $\delta\ell$ steps) along the transition path does not increase much the energy, and the transition process is fast compared to activated hopping between deep minima taking place in the Hopfield model. Studying quantitatively the Monte Carlo dynamics of Random-RBM model is an interesting lead for future work.

4 Available at `http://www.phys.ens.fr/~monasson/papers.html`

8

QUANTITATIVE COMPARISON WITH RBM TRAINED ON MNIST

In this chapter, we compare theoretical predictions from the Random-RBM ensemble and real RBM trained on MNIST. We first derive numerical proxies for the model parameters (p, β, g, θ,$\rho(x)$), then show results.

8.1 FINDING ATTRACTORS IN RBM TRAINED ON MNIST

In RBM trained on real data, an attractor is defined as a local maximum of the marginal distribution $P(\mathbf{v})$. Importantly, the attractors of $P(\mathbf{v})$ are not necessarily the attractors of $P(\mathbf{v}, \mathbf{h})$; this is because at finite temperature, the entropy of the hidden layer matters. Though sampling from $P_\beta(\mathbf{v})$ is not possible in general, it can be done for integer β, see Section 13.1.2. In particular, when $\beta \to \infty$, it can be shown that the zero temperature sampling Gibbs step is given by:

$$\begin{aligned} h_\mu &\leftarrow \mathbb{E}\left[h_\mu | v\right] \\ v_i &\leftarrow \Theta\left[g_i + \sum_\mu w_{i\mu} h_\mu\right] \end{aligned} \tag{8.1}$$

It is in general very difficult to enumerate all the attractors of the model, as there may be an exponential number of them. Here, we computed a subset of attractors for Figures 3.12 and 8.3 by starting from the train set configurations, and performing zero temperature sampling until convergence.

8.2 NUMERICAL PROXIES FOR CONTROL AND ORDER PARAMETERS

Several control and order parameters are well defined for R-RBM in the thermodynamical limit, but not for typical RBM trained on data. For R-RBM instances,

the average weight sparsity p is well defined because the weights take only three distinct values $\{-\frac{1}{\sqrt{N}}, 0, \frac{1}{\sqrt{N}}\}$, but for RBM trained on data, the weights $w_{i\mu}$ are never exactly equal to zero. Similarly, the number of strongly activated hidden units L is well-defined for R-RBM in the thermodynamic limit $N \to \infty$ because their activity scales as $\sqrt{N}$; but in general, all hidden units have finite activation. Proxies are therefore required to compare theoretical and numerical results. We consider 'consistent' proxies, giving back (in the large size limit), the original parameters for RBMs drawn from the R-RBM ensemble.

8.2.1 *Participation Ratios PR*

Participation ratios are used to estimate numbers of nonzero components in a vector, while avoiding the use of arbitrary thresholds. The Participation Ratio (PR_a) of a vector $\mathbf{x} = \{x_i\}$ is

$$PR_a(\mathbf{x}) = \frac{(\sum_{\mathbf{i}} |\mathbf{x_i}|^{\mathbf{a}})^{\mathbf{2}}}{\sum_{\mathbf{i}} |\mathbf{x_i}|^{\mathbf{2a}}}$$

If $\mathbf{x}$ has K nonzero and equal (in modulus) components PR is equal to K for any a. In practice we use the values $a = 2$ and 3: the higher a is, the more small components are discounted against strong components in $\mathbf{x}$.

8.2.2 *Number L of active hidden units*

In both numerical simulations of R-RBM and on RBM trained on MNIST, we estimate L, for a given hidden-unit configuration $\mathbf{h}$, through

$$\hat{L} = PR_3(\mathbf{h})$$

To understand the choice $a = 3$, consider a typical activation configuration $\mathbf{h}$ for a R-RBM :

$$h_\mu = \begin{cases} m\sqrt{N} & \text{if} \quad 1 \leq \mu \leq L \,, \\ \sqrt{r}\, x_\mu & \text{if} \quad L+1 \leq \mu \leq M \,, \end{cases} \tag{8.2}$$

where the magnetization m and mean square activity r are $\mathcal{O}(1)$, and x_μ are random variables with zero mean, and even moments of the order of unity. The first L hidden units are strongly activated ($\mathcal{O}(\sqrt{N})$ activity), whereas the

remaining $N - L$ others are not (activations of the order of 1). Here, we assume L to be finite as $N \to \infty$. One computes :

$$PR_2(h) \sim \frac{(Lm^2N + (N-L)r)^2}{Lm^4N^2 + (N-L)r^2} = L \times \frac{(1 + \frac{N-L}{N}\frac{r}{Lm^2})^2}{1 + \frac{N-L}{N^2}\frac{r^2}{Lm^4}} \underset{N\to\infty}{\longrightarrow} L(1 + \frac{r}{Lm^2})^2 ,$$
$$PR_3(h) \sim \frac{(Lm^3N^{3/2} + (N-L)r^{3/2})^2}{Lm^6N^3 + (N-L)r^3} = L \times \frac{(1 + \frac{N-L}{N^{3/2}}\frac{r^{3/2}}{Lm^3})^2}{1 + \frac{N-L}{N^3}\frac{r^3}{Lm^6}} \underset{N\to\infty}{\longrightarrow} L . \tag{8.3}$$

Hence choosing coefficient $a = 3$ ensures that the participation ratio (a) does not take into account the weak activations in the thermodynamical limit, and (b) converges to the true value L if all magnetizations are equal.

8.2.3 *Normalized Magnetizations*

Given a RBM and a visible layer configuration, we define the normalized magnetization of hidden unit μ as the normalized overlap between the configuration and the weights attached to the unit:

$$\tilde{m}_\mu = \frac{\sum_i (2v_i - 1)w_{i\mu}}{\sum_i |w_{i\mu}|} \in [-1, 1]$$

This definition is consistent with the Hopfield model. For R-RBM, we also have, in the thermodynamical limit, $\hat{m}_\mu = \frac{2I_\mu}{p\sqrt{N}}$, where I_μ is the input received by the hidden unit from the visible layer; m_μ is $\mathcal{O}(1)$ for strongly activated hidden units, and $\mathcal{O}(\frac{1}{\sqrt{N}})$ for the others.

For a given configuration $\mathbf{v}$, with $\hat{L}$ activated hidden units, the normalized magnetization of the activated hidden units $\tilde{m} = \frac{m}{p/2}$ can be estimated as the average of the $\hat{L}$ highest magnetizations $\hat{m}_\mu$.

8.2.4 *Weight sparsity p*

A natural way to estimate the fraction of non-zero weights $w_{i\mu}$ would be to count the number of weights with absolute value above some threshold t. However, there is no simple satisfactory choice for t. Indeed, the fraction of non-zero weights should not depend on the scale of the weights, i.e. it should be invariant under the global rescaling transformation $\{w_{i\mu}\} \to \{\lambda\, w_{i\mu}\}$. As the scale of

weights vary from RBMs to RBMs and, for each RBM, across training it appears difficult to select an appropriate value for t. A possibility would be to use a threshold adapted to each RBM, e.g. $t \propto \kappa\sqrt{\frac{W_2}{M}}$, where κ would be some small number. Our experiments show that it is not accurate enough, due to the scale disparities across the hidden–unit weight vectors $\mathbf{w}_\mu$. Rather than adapting thresholds to each hidden unit of each RBM, we use Participation Ratios, which naturally enjoy the scale invariance property. We estimate the fraction of nonzero weights through

$$\hat{p} = \frac{1}{MN}\sum_\mu PR_2(\mathbf{w}_\mu)$$

For R-RBM with $w_{i\mu} \in [-W_0, 0, W_0]$ with corresponding probabilities $[\frac{p}{2}, 1-p, \frac{p}{2}]$, the estimator is consistent: $\hat{p} = p$.

Other consistent estimators for p are possible, such as averaging the fraction of nonzero weights for a given visible unit, $p_i = \frac{PR_2(\mathbf{w}_i)}{M}$. For RBMs trained on MNIST, they typically have similar numerical values.

8.2.5 *Weights heterogeneities*

Not all visible units are equally connected to the hidden layer. To better capture this effect, one can study R-RBM with any arbitrary distribution of p_i. Analogously to the homogeneous case a high sparsity limit is obtained when the average sparsity, $p = \frac{1}{N}\sum_i p_i$, vanishes. We define the distribution of the ratios $\tilde{p}_i = \frac{p_i}{p}$ in the $p \to 0$ limit. In practice the ratios are estimated through

$$\tilde{p}_i = \frac{\sum_i w_{i\mu}^2}{\frac{1}{M}\sum_{i,\mu} w_{i\mu}^2} \,. \tag{8.4}$$

For a heterogeneous R-RBM, we have consistently $\tilde{p}_i = \frac{\hat{p}_i}{p} = \frac{p_i}{p}$. Looking at the histogram of values of $\tilde{p}_i$ across all RBM inferred on MNIST, we find a non-negligible spread around one, see Fig. 8.1. We also display for each visible unit i the average of $\tilde{p}_i$ accross all RBM inferred; we can see that the visible units at the border are indeed the least connected (smaller $\tilde{p}_i$), whereas the ones at the center are strongly connected (larger $\tilde{p}_i$).

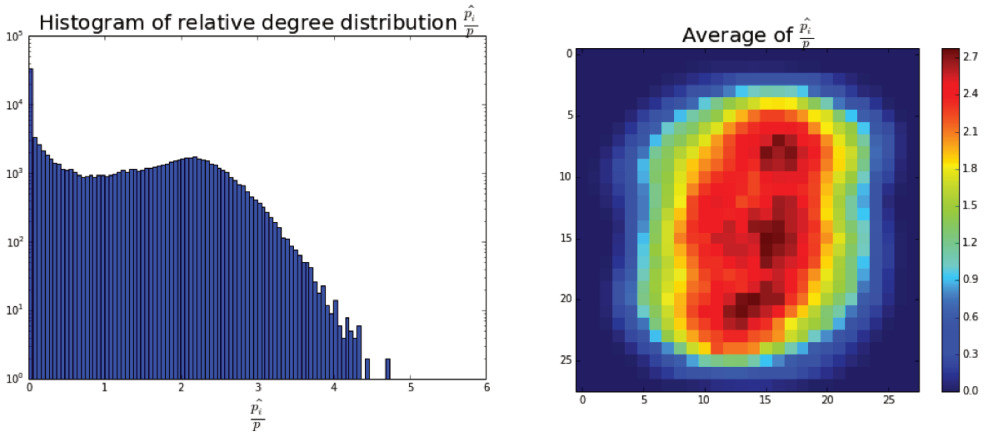

Figure 8.1: (a) Histogram of $\tilde{p}_i = \frac{p_i}{p}$ values, across all visible units and RBMs inferred on MNIST. (b) Average across all RBM of $\tilde{p}_i = \frac{p_i}{p}$, for each visible unit

8.2.6 *Effective Temperature T*

Although RBM distributions are always defined at temperature $T = 1$, the effective temperature is not 1. This is very much like in the Ising model : the behavior of the system depends on an effective temperature $\hat{T} = \frac{T}{J}$ where J is the coupling strength; a low effective temperature phase correpond to high values of J. For ReLU RBM, the probability distribution of configurations at temperature T is defined as :

$$P_{\mathbf{w}}[\mathbf{v},\mathbf{h}] = e^{-\frac{E[\mathbf{v},\mathbf{h}]}{T}} \quad \text{with} \quad \frac{E[\mathbf{v},\mathbf{h}]}{T} = -\sum_i \frac{g_i}{T} v_i + \sum_\mu \left(\frac{h_\mu^2}{2T} + \frac{h_\mu \theta_\mu}{T} \right) - \sum_{i,\mu} \frac{w_{i\mu}}{T} v_i h_\mu \,. \tag{8.5}$$

Let $\bar{\mathbf{h}} = \frac{\mathbf{h}}{\sqrt{T}}$. The probability can be rewritten as $P[\mathbf{v},\bar{\mathbf{h}}] = e^{-\bar{E}[\mathbf{v},\bar{\mathbf{h}}]}$ with

$$\bar{E}(\mathbf{v},\bar{\mathbf{h}}) = -\sum_i \frac{g_i}{T} v_i + \sum_\mu \left(\frac{\bar{h}_\mu^2}{2} + \bar{h}_\mu \frac{\theta_\mu}{\sqrt{T}} \right) - \sum_{i,\mu} \frac{w_{i\mu}}{\sqrt{T}} v_i \bar{h}_\mu \,. \tag{8.6}$$

Since the marginal $P[\mathbf{v}]$ is not affected by the change of variable, a ReLU RBM at temperature T is therefore equivalent to another ReLU RBM at temperature $T = 1$, with new fields, thresholds and weights : $\bar{\mathbf{g}} = \frac{\mathbf{g}}{T}$, $\bar{\theta} = \frac{\theta}{\sqrt{T}}$, $\bar{\mathbf{w}} = \frac{\mathbf{w}}{\sqrt{T}}$. Therefore, changing the temperature is equivalent to rescaling the parameters,

and in turn, the effective temperature of a given RBM can be deduced from the amplitude of its weights. For a R-RBM at temperature T:

$$W_2 = \frac{1}{M}\sum_{\mu,i} \bar{w}_{i\mu}^2 \underset{N\to\infty}{\sim} \frac{p}{T} \, .$$

We therefore estimate the temperature of a given RBM through

$$\hat{T} = \frac{\hat{p}}{\frac{1}{M}\sum_{\mu,i} w_{i\mu}^2} \, .$$

From this definition, it can be seen that the low temperature regime of the compositional regime, $T \ll p$, is equivalent to $W_2 \gg 1$. In RBM trained on MNIST, we typically find $W_2 \sim 7$

8.2.7 *Fields g*

Similarly to the weights, the fields g_i and normalized fields could be estimated respectively as:

$$\begin{aligned} \hat{g}_i &= \hat{T}\bar{g}_i \\ \hat{\tilde{g}}_i &= \frac{\hat{T}}{\hat{p}}\bar{g}_i = \frac{\bar{g}_i}{\frac{1}{M}\sum_{\mu,i} w_{i\mu}^2} \end{aligned} \tag{8.7}$$

A naive estimate for the normalized field $\tilde{g}$ would be to average the fields: $\hat{\tilde{g}} = \frac{1}{N}\sum_i \hat{\tilde{g}}_i$. It is however not really meaningful, as the $\hat{\tilde{g}}_i$ are extremely heterogeneous: for instance, the mean value over the sites i of a single RBM is equal to -0.48, and is comparable to the standard deviation, 0.40. This range of variation spans all the phases of R-RBM. To achieve quantitative predictions, we instead adjust the R-RBM parameter g so that q, the mean value of v_i in the visible layer, averaged over thermal fluctuations and quenched disorder, matches the value 0.132 obtained from MNIST data. This gives $\frac{\hat{\tilde{g}}}{\hat{p}} = -0.1725$ for homogeneous R-RBM, and $\frac{\hat{\tilde{g}}}{\hat{p}} = -0.21$ for heterogeneous R-RBM.

8.2.8 *Thresholds θ*

The thresholds and normalized thresholds can be estimated as

$$\begin{aligned}\hat{\theta}_\mu &= \sqrt{\hat{T}}\,\bar{\theta}_\mu \\ \hat{\tilde{\theta}}_\mu &= \sqrt{\frac{\hat{T}}{\hat{p}}}\,\bar{\theta}_\mu = \frac{\bar{\theta}_\mu}{\sqrt{\frac{1}{M}\sum_{\mu,i} w_{i\mu}^2}}\end{aligned} \tag{8.8}$$

Again, a naive estimate for the normalized threshold $\tilde{\theta}$ would be the average $\hat{\tilde{\theta}} = \frac{1}{M}\sum_\mu \hat{\tilde{\theta}}_\mu$ but this estimate is not meaningful. Indeed, contrary to the R-RBM case, the inputs I_μ of the hidden units μ are not evenly distributed around zero: $\mathbb{E}\left[I_\mu\right] \neq 0$. Hence, even if the threshold is equal to zero, the activation probability can be different from 0.5. We take this effect into account by substracting the average value of the inputs from the average of θ, and find that the difference is equal to 0.33, with standard deviation 1.11. This range of value for θ again spans all phases. In order to use a well-defined value, we choose θ such that the critical capacity $\alpha_c^{R-RBM}(\ell_{max}) = 0.5$, where $\ell_{max} \sim 1.5$ is the maximum average index number observed across all RBMs trained. This estimation gives $\hat{\tilde{\theta}} \sim 1.5$.

8.3 RESULTS

We first evaluate the scaling law $L \propto \frac{\ell}{p}$. Compared to Fig. 3.8, we add a regularization penalty $\propto \sum_\mu (\sum_i |w_{i\mu}|)^x$ to control the final degree of sparsity; the case $x = 1$ gives standard L_1 regularization, while, for $x > 1$, the effective penalty strength $\propto (\sum_i |w_{i\mu}|)^{x-1}$ increases with the weights, hence promoting homogeneity among hidden units. After training we generate Monte Carlo samples of each RBM at equilibrium, and monitor the average number of active hidden units, $\hat{L}$, and the normalized magnetization, $\tilde{m}$. Figure 8.2(a) shows $\hat{L}$ vs. $\hat{p}$, in good agreement with the R-RBM theoretical scaling $L \sim \frac{\ell^\star}{p}$. Figure 8.2(b) shows that $\tilde{m}$ is a decreasing function of $\ell = \hat{L} \times \hat{p}$, as qualitatively predicted by theory, but quantitatively differs from the prediction of R-RBM with homogeneous p. This disagreement can be partly explained by the heterogeneities in the sparsities p_i in RBMs trained on MNIST, e.g. units on the borders are connected to only few hidden units, whereas units at the center of the grid are connected to many. Using the empirical heterogeneous degree distribution $\rho(x)$ yields improved fit accuracy for both $\ell^\star$ and $\tilde{m}$.

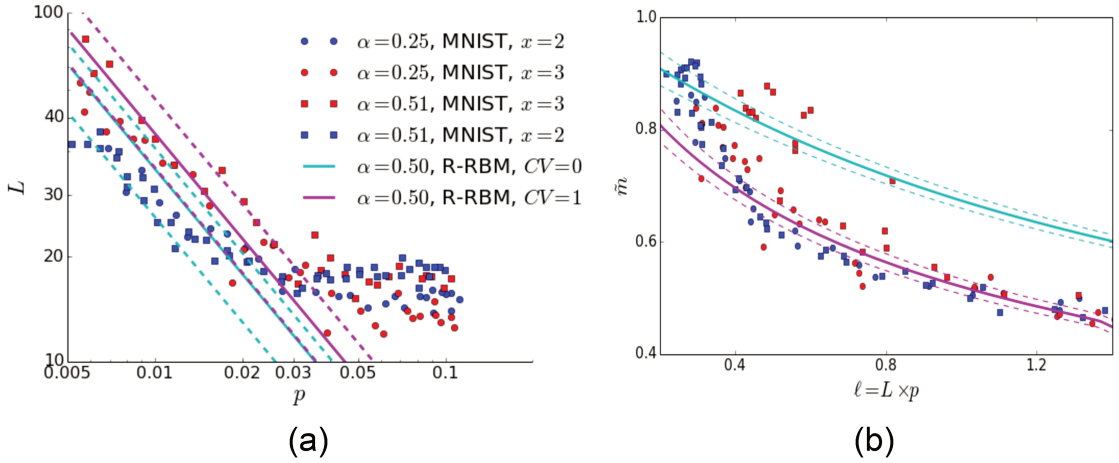

Figure 8.2: **Quantitative predictions in the compositional regime of R-RBM compared to RBMs inferred on MNIST**. Each point represent a ReLU RBM trained with various regularizations, yielding different weight sparsities p. Solid lines depict predictions found by minimizing f_ℓ, and dashed line expected fluctuations at finite size (N) and temperature. **(a)** Average number L of active hidden units vs. p. **(b)** Average magnetization $\tilde{m}$ vs. $\ell = L \times p$.

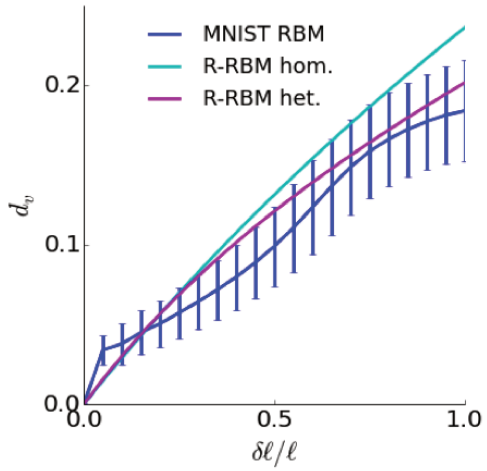

Figure 8.3: Distance (per pixel) between pairs of visible attractors vs. relative distances in the hidden-unit activation patterns. Purple and cyan lines are theoretical R-RBM predictions from Eqn. (7.38). Blue lines denote mean and standard deviations for pairs of attractors of RBM inferred from MNIST. For MNIST RBM, $\delta\ell/\ell$ is computed using participation ratios as $\hat{L}$.

DISCUSSION

ReLU and Bernoulli RBM were shown empirically to be efficient as feature extraction algorithm, as well as good generative models. By studying an ensemble of RBM with random weights, we found theoretical insights consistent with these observations. The combination of sparse weights, low effective temperature, fields and non-linearities allows to drive RBM in a compositional phase, in which i) typical visible layer configurations drawn from the model have a simple hidden layer representation, with a few strongly activated hidden units ii) the probability $P(\mathbf{v})$ is very rough, with a large diversity of local maxima arranged in a specific geometry. The phenomenology of Random-RBM matches well the one of RBM trained on real data, both qualitatively and quantitatively.

Beyond quantitative modeling, the compositional phase refines our understanding of how and why RBM work well:

- RBM are good feature extractors because in the compositional phase, there is a simple relationship between the typical configurations from $P(\mathbf{v})$ and the weight matrix; and therefore between the real data used from training and the weights. Configurations are essentially generated by recombining the extracted features.

- RBM are good generative models because they can produce a large diversity of 'well-formed' (*i.e.* not noisy) configurations. In particular, the ability of RBM to generate configurations that are significantly different from the ones in the training set arises directly from this compositionality: high-probability attractors can be obtained by recombining features in a way that was unseen in the training data.

- Higher-order (i.e. non-linear) RBM outperform pairwise model because the non-linearity prevents the cross-talks between the hidden units, which can severely impair performance.

- In a compositional phase, one can transit from one attractor to the other by gradual changes of the hidden layer representation. Though a quantitative analysis is clearly desired, this may explain why RBM MCMC mix well in practice, and PCD is good enough.

Our work may be challenged both from a technical and a conceptual point of view. First, several technical shortcuts were taken:

- We used a simple uniform Ansatz for the magnetized solution $m_\mu = m \forall \mu \leq L$, despite the fact that the real ground state is in general non-uniform, as seen from the case $\alpha = 0$. Differences of order p are expected.

- We assumed a replica symmetric Ansatz, but it is known to be inexact at low temperature both for the Hopfield and SK models. Although quantitative differences are very small for the Hopfield model in practice, we acknowledge that a more cautious computation is required.

- A combinatorial diversity of solutions of the free energy minimization was found, but we have not checked how many of them are significantly contributing at finite temperature. In particular, it would be important to compute the average entropy $S(T)$ so as to estimate an effective number of possible combinations.

Secondly, the scope of the computation is limited to simple cases. Random-RBM are a fairly crude model of real RBM: for instance, weights are not randomly distributed but often concentrated in regions, such that the overlaps between the patterns or their supports are not uniformly identical $\sim p^2$ for all hidden units pairs. Patterns overlaps, which are also missing, are notably important for explaining correlations between hidden units. Though we acknowledge this limitation, we also argue that we are not trying to design an accurate model of RBM learnt from MNIST specifically, but rather to learn general properties of RBM trained on data. Moreover, the computation requires $\mathcal{U}(h) \sim h^2$, and therefore it does not generalize to Bernoulli hidden units. Indeed, a ferromagnetic or compositional regime may be observed only if one or few hidden units can dominate over the $\sim \alpha N \to \infty$ others; and therefore h must be unbounded. With Bernoulli hidden units, we speculate two different scenarios: either $p \sim \frac{1}{N}$, such that each hidden unit is effectively always in a finite N regime, or $p \sim 0.1$ and there is a 'copy' mechanism, such that several hidden units share the same weights $w_{i\mu}$. Indeed, Nair and Hinton have shown that duplicating Bernoulli hidden units with identical weights and varying thresholds effectively resulted in a single ReLU-like hidden unit [93]; Bernoulli RBM may therefore work in practice just like ReLU RBM, provided the features are strongly overlapping. In our experiments on proteins, we have observed larger overlaps between hidden unit patterns for Bernoulli hidden units than Gaussian or ReLU, which is consistent with this hypothesis.

We have not studied explicitly the training dynamics in maximum likelihood. It would be interesting to study in particular the observed transition between

Prototypic and Compositional representation. Moreover, it would also be important to understand why in some cases weight sparsity naturally arise from maximum likelihood as in MNIST and in others it does not, as in proteins (see next part). Since the publication of our paper, two works aimed at better understanding the dynamic of learning in RBM were recently published [144, 145]; future work in this direction are welcome. Generalizations of our computation to deep models is also another important study. Nethertheless, regardless of the training procedure (maximum likelihood or other principle, regularization,...), a RBM will behave in a compositional phase provided that the weights are sparse. We will heavily rely on this property for modeling RBM trained on protein sequences by enforcing sparsity through regularization.

Part IV

MODELING PROTEIN SEQUENCES WITH RESTRICTED BOLTZMANN MACHINES

9

BACKGROUND

9.1 CONTEXT

Proteins form the basis of life. Proteins are large macromolecules constituted by sequences of amino-acids linked together by peptidic bonds. After transcription from the DNA, the protein, initially in a linear conformation, folds into a sequence-specific three-dimensional structure that defines its functional properties. Its unique shape allows the protein to interact selectively with other molecules or proteins with complementary shape in order to perform various tasks, such as catalyzing metabolic reactions, detecting biochemical stimuli, structuring the cell, transporting molecules,... Beyond this simple picture and despite decades of intensive research in structural biology, computational protein modeling and bioinformatics, we still have a limited mastery over the relationship between the sequence, structure and function of proteins. Indeed, several factors seriously limit our ability to emulate the physical processes underlying proteins life.

First and foremost, the space of possible proteins sequences is huge: with 20 different amino-acids, there are $\sim 10^{130}$ possible sequence of length 100, which clearly makes exhaustive experimental characterization of all proteins impossible. Though impressive, this number would not be a barrier if only there were simple symmetry or continuity properties of the function mapping sequence to structure / function: after all, there is an infinity of possible electric charge and current spatial distributions, yet we can predict very accurately the induced electromagnetic field thanks to the Maxwell equations. This is because satisfying the constraints of space-time symmetries and continuity (in the sense of the Euclidian distance) leave very few candidate mappings; yielding relatively easy identification and very good predictive power. Unfortunately, the picture is not so simple for proteins, because high sequence identity does not necessarily imply structural similarity. On the one hand, two proteins, e.g. Kunitz domains from a bacterial (resp. eukaryote) organism may have as few as 20% sequence identity, yet have almost identical structure and function. On the other hand, a single mutation of amino-acid may completely impair a protein's ability to fold and function properly, resulting in potentially deadly genetic

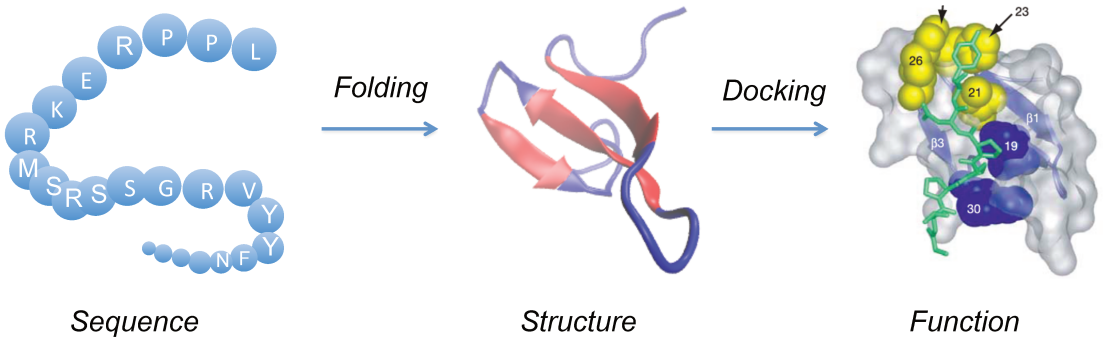

Figure 9.1: The Sequence - Structure - Function relationship for the WW domain: A given sequence, initially produced in a linear conformation folds into a specific structure, which allows it to interact with target ligands. PDB Structure: 1eom [146]. Right panel from [147]

disorder for the carrier. Finding suitable metrics, as well as efficient ways to navigate through the space of functional sequences is therefore a fundamental challenge of bioinformatics.

Secondly, each of these sequences has many degrees of freedom (namely the set of torsion angles from one amino-acid to the next), yielding a very large numbers of possible spatial configurations. As stated in the Levinthal's paradox, finding the native conformation of a 100 length sequence by exploring each of these configurations at the speed of light would take 10^{75} years [148]. Fortunately for us, natural folding is guided by free energy decrease and takes only of the order of few millisecond [149]. Unfortunately for us, it is very difficult to emulate this process using numerical simulations. Indeed, an accurate Ab Initio Molecular Dynamics requires modeling each atom of the protein, plus dozens of solvent molecules which are key of hydrophobic interactions, on a temporal scale from pico-seconds to milliseconds. Overall, this adds up to a tremendous computational cost, and it takes a specially designed Anton Supercomputer to fold a protein with 80 residues [150]. Coarse grained models, which simulate the protein at the level of amino-acids using knowledge-based potentials are less costly, but not as accurate and can fail to find the native structure for large proteins [151, 152].

Lastly, it is difficult - though not as hard as folding - to predict whether and how a folded protein will bind to a target molecule or another protein. Docking algorithm such as Haddock and Patchdock [153, 154] can evaluate complementarity of two protein surfaces by scanning through the different relative positions of two rigid structures and estimating their interaction energy using knowledge-based potentials. It is particularly powerful for drug-design applications, but remains limited in practice by conceptual flaws: the structure

of a protein in complex often differs from its unbound structure, either due to conformation selection (i.e. another ground state is selected through the interaction) or induced fit (the ground state is perturbed by the interaction) [155]. In practice, this can result in a large number of false positive solutions for docking.

These limitations have several very concrete practical consequences. Because we cannot predict systematically and accurately protein structures and complexes from their sequences only, it can be very hard to elucidate the behavior of complex protein networks. A famous example is the case of the Alzheimer disease, in which amyloid β protein aggregates form in the brain, causing neurons death and neurodegenerescence [156]. To this day, it remains unclear what is the cascade of molecular malfunction that triggers the accumulation, or even what is the original functional role of the Alzheimer Precursor Protein (APP, the protein that degenerates into amyloid plaques) in the first place [157].

Even for well-understood diseases, our practical ability to design drugs is also limited. Current docking algorithm are often good enough to design compounds that can bind very well to a target protein and inhibit its interactions with other proteins [158, 159]. However, because conformational changes within protein complexes are not well understood, it is very difficult to design drugs that, when complexed with a target malfunctional protein, can restore its original structure and functionality [160].

Finally, the ability to design artificial proteins with desired properties remains limited. The most commonly used protein strategy is Directed Evolution (DE), which mimics *in vitro* natural evolution cycle. Starting from an initial protein, a library of variants -each with a few mutations/insertions/deletions - is created, and these variants are expressed in organisms, e.g. on the surface of phages [161]). The library is then subjected to functional selection: for instance, a target substrate is set on a solid support, and only the phages expressing a protein that strongly binds to the substrate stick on the support. The surviving phages are then amplified, and another cycle can be performed. DE techniques notably led to the development of therapeutic monoclonal antibodies, and were recently awarded the 2018 Chemistry Nobel Prize. The main limitation of DE is that it can only explore a limited region of the sequence space around the initial sequence, and therefore, one must start from an initial sequence that is already functional. This limits DE to the optimization of protein function (binding affinity, solvability, thermal stability,...) rather than to the development of radically new functionality.

Computational methods of *Ab initio* protein design have the potential to better explore the sequence space and have found interesting successes, but

they are also limited in practice. Initial computational design were based on so-called threading algorithm, which are based on heuristic free energy functions that assesses whether a sequence is well fit to a given fold. The first successful *Ab initio* protein design simulations reported small artificial sequences whose structure was very close to the target natural [162] or artificial fold [163]. However, as for folding and docking, these heuristic often fail to take into account important effects, such as the role of competing folds. In particular, designing flexible protein parts such as loops, which are often key to binding function is challenging. Lastly, threading-based design of sequences with a target fold and high-binding affinity with some substrate requires knowledge of the protein complex, which is not always available. More recently, computational approaches using Molecular Dynamics or frameworks such as Rosetta have found encouraging successes [164–166] but designing new, large, folds remains exceptional [167, 168]. Developing principled, accurate and affordable protein design strategies could lead to giant leaps for both basic research and industrial applications. For instance, new fluorescence calcium indicators similar to GCaMP [169] could help probing large biological neural networks at shorter time scales, whereas artificial ion channels could lead to new filtering water systems as efficient as our cell's.

In parallel to these developments, the last two decades have witnessed tremendous improvements in DNA sequencing techniques. As a consequence, the number of available protein sequences has exploded: there are currently ~ 120 millions sequences on the UniprotKB database, of which only about 0.5% are annotated and have a known function (SwissProt database [170]). This raises both important challenges and opportunities. On the one hand, we do not know what do most of these proteins do at all and it is crucial to develop automated structure and function prediction tools. On the other hand, the statistics of these sequences carry information about their underlying structure and function. For instance, amino-acid conservation suggests structural or functional importance, whereas correlation between amino-acid mutations can indicate proximity on the 3D structure. The main goal of coevolution is to develop systematic methods to unveil such properties. As such, coevolution lies at the interface between Bioinformatics, Statistical Physics and unsupervised Machine Learning, and our thesis takes place in this context. Important improvements over traditional methods of structural and functional predictions were recently brought by including coevolution forecasts. They will be reviewed briefly in section A.

9.2 COEVOLUTION

9.2.1 *Natural Selection and conservation*

The starting point of coevolution methods is the observation that all existing natural protein sequences are good at something. Simply said, if one protein sequence was not doing its job correctly - whatever it is - its host would die and we would never have seen this sequence ! Conversely, 5 billion years of evolution could not have left a useless protein in the genome of a modern organism. As an introductory example, consider the following set of sequences of the WW domain shown in Fig. 9.2, which have identical structure and function. As highlighted, essentially all sequences carry a Tryptophan (W) at positions 5 and 29. Or - to rephrase it - we never see a sequence that does not carry a Tryptophan. Clearly, over the billions of years of existence of the WW domain, mutations of these sites have arised many times, yet we do not see it. This makes sense only if W5, W29 are crucial for function, such that sequences carrying these mutations were selected away by evolution. As a matter of fact, experimental structures of the WW domain have indeed shown that these two sites are the main binding sites of the WW domain, i.e. sites in direct proximity with the ligand (target molecule) and directly responsible for the complex formation. In order to go beyond this simple evolutionary pattern, a few notions are required:

- *A*mino-acids The full list of the 20 amino-acids is available on Table 9.1 and their properties are visualized on Fig. 9.3. Amino-acids differ by their side chain chemical properties, including size, electrical charge, aromaticity, hydrophobicity. Some amino-acids such as Isoleucine (I) and Leucine (L) are very similar and are often exchangeable within a sequence, whereas others are radically different, such as Aspartic Acid (D, acidic, negatively charged) and Lysine (K, basic, positively charged). For visualization purposes, we have divided them in 8 subgroups based on their properties, and assigned a color to each.

- *M*SA A Multiple Sequence Alignment (MSA) is a way of arranging the sequences of proteins to identify similarities and differences between them. Two evolutionary-related sequences may differ by substitutions at selected sites, but also by insertions and deletions of sites. To account for this, we build an MSA by identifying and aligning the matching sites across the various proteins, and treat the others residues as insertions; missing sites due to a deletion are represented by the gap symbol (-). The final result is a matrix where each row is a sequence, each column

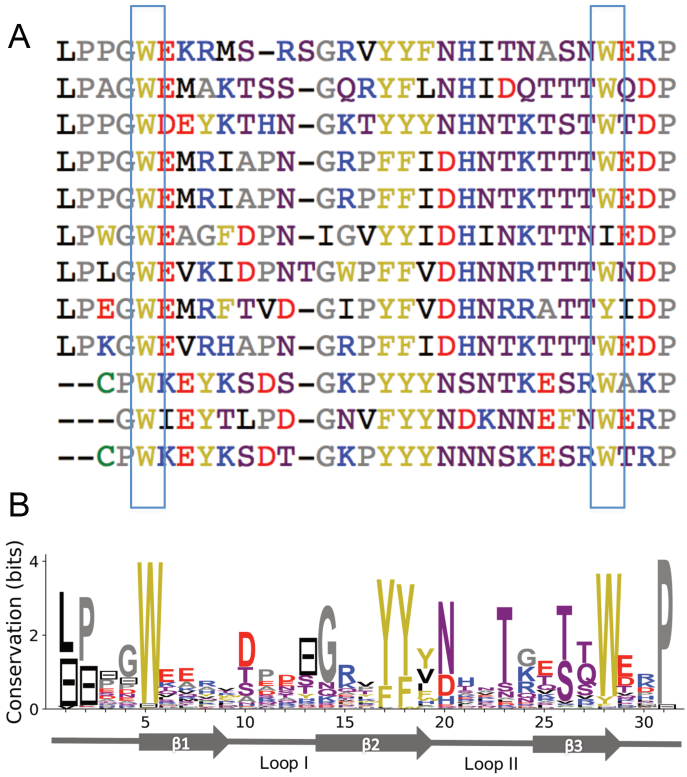

Figure 9.2: A. Multiple Sequence Alignment of the WW Domain. Rectangles highlight conserved sites. B. Corresponding Sequence Logo visualization

is a site and each entry either one of the 20 amino-acids or a gap, see Fig. 9.2. Building an MSA from a given set of sequences amounts to finding a sequence correspondence that maximizes homology between sequences while minimizing the number the of gaps; such optimization is implemented in practice using dynamic programming such as the Needleman-Wunsch algorithm.

- Conservation Score Let $f_i(a)$ be the observed frequency of each of the 20+1 amino-acids a, and S_i its corresponding Shannon entropy: $S_i = -\sum_a f_i(a) \log f_i(a)$. Then the conservation score of a site i is :

$$C_i = \log 21 - S_i \tag{9.1}$$

  By construction, $C_i = 0$ if a site is completely unconserved, *i.e.* when all amino-acid plus gap have identical frequency $\frac{1}{21}$, and $C_i = \log 21 \sim 4.4$bits when it is completely conserved.

- A Sequence logo, such as the one shown in Fig. 9.2, is a standard data visualization of the pattern of conservation within a MSA. Each column represents a site, with total height equal to C_i; it is filled with letters representing the amino-acids, sorted by frequency (top ones are most frequents) and with height proportional to $f_i(a)$.

Here, the Sequence Logo of the WW-domain MSA shown in Fig. 9.2 allows us to find more complex patterns of conservation. For instance, sites 17 and 18 are not perfectly conserved, but only two amino-acids (Tyrosine or Phenylalanine) are possible, both of which are aromatic. Similarly, sites 23, 26 and 27 are almost always occupied by polar hydrophilic residues and never by hydrophobic residues, suggesting that this site may often be in contact with water molecules of the solvent. More broadly, it is the combination of the various structural and functional chemical constraints, such as solvent exposure, steric interactions, surrounding charge... that determines which amino-acids can be present at a given site of a given sequence.

Beyond conservation, such structural and functional constraints also induce non-trivial second order moments of the amino-acid distribution. In particular, sites that are far away on the sequence but close in the tertiary structure of the protein can undergo coevolution, *i.e.* correlations between mutations. For instance, if the first residue is positively charged (H,K,R), then the second cannot be positively charged as well, because eletrostatic repulsions would destabilize the structure; instead, negatively charged residues (D,E) may be observed. In other sequences of the alignment, the first residue may itself be negative, and conversely positively charged residues are favored on the

Alanine	Ala	A	Leucine	Leu	L
Arginine	Arg	R	Lysine	Lys	K
Asparagine	Asn	N	Methionine	Meth	M
Aspartic Acid	Asp	D	Phenylalanine	Phe	F
Cysteine	Cys	C	Proline	Pro	P
Glutamic Acid	Glu	E	Serine	Ser	S
Glutamine	Gln	Q	Threonine	Thr	T
Glycine	Gly	G	Tryptophan	Trp	W
Histidine	His	H	Tyrosine	Tyr	Y
Isoleucine	Ile	I	Valine	Val	V

Table 9.1: List of Amino-Acids and their abbreviations

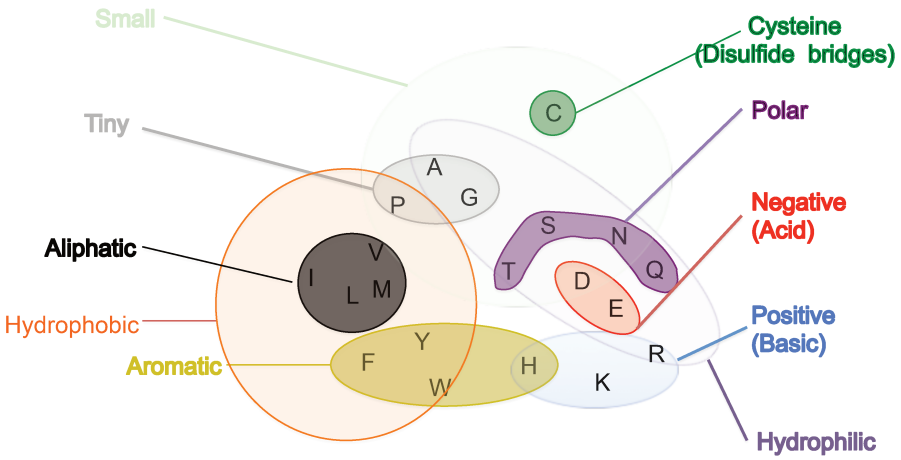

Figure 9.3: Venn Diagram summarizing the amino-acid properties. The following color code is used hereafter: red = negative charge (E,D), blue = positive charge (H, K, R), purple = non-charged polar (hydrophilic) (N, T, S, Q), yellow = aromatic (F, W, Y), black = aliphatic hydrophobic (I, L, M, V), green = cysteine (C), grey = other, tiny (A, G, P).

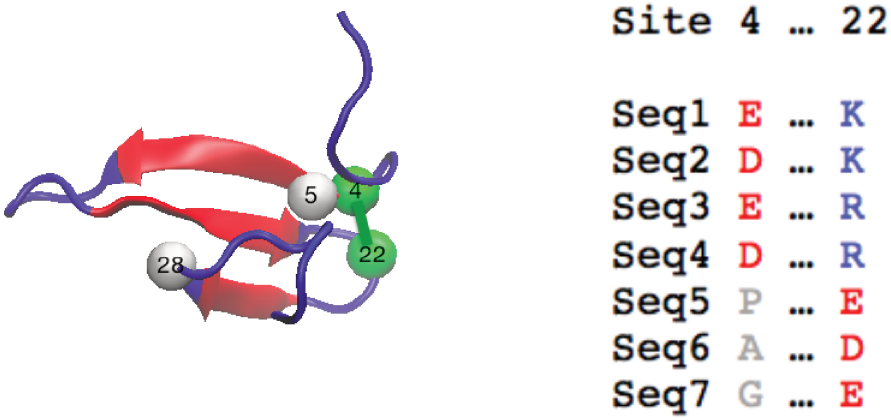

Figure 9.4: Two coevolving sites on the WW domain

second site. Statistically, we would thus observe a negative correlation, i.e. $C_{ij}(a,b) \equiv f_{ij}(a,b) - f_i(a)f_j(b) << 0$ for $a, b \in \{D, E\}$, and sometimes a positive correlations for $C_{ij}(a,b) > 0$ for $a \in \{D, E\}$, $b \in \{H, R, K\}$ and vice-versa. In the example of the WW domain, we measure negative correlations between negatively charged residues of sites 4 and 22, which are in contact, see Fig. 9.4.

Can we exploit this signal to predict whether two sites are in contact or not ? In principle, this is possible. Starting from a sequence or a group of sequences with similar structure, we can build a large MSA by looking for homologous sequences with presumed similar structure and function in a sequence data basis (e.g. UniprotKB) using BLAST or Hidden-Markov Models such as HMMER [171]. This MSA can then be used to compute various coevolution scores such as the mutual information for each pair of sites [172]. In practice, predicting contacts from such pairwise metrics yields a large number of false positive predictions. Besides, even more sophisticated metrics based on supervised learning approaches [173] give quite low performance. The main difficulty lies in separating efficiently this coevolution signal from the other sources of correlations in the data-set, such as phylogenic and indirect correlations. About 10 years ago, graph-based approaches such as the Direct Coupling Analysis (DCA), briefly introduced below, have brought tremendous improvements to coevolution-based contact predictions. Coevolution is now routinely used in several structure and function prediction pipelines.

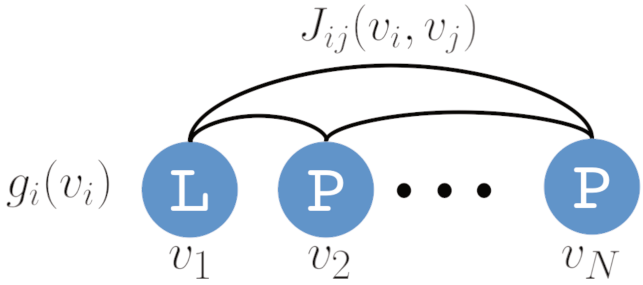

Figure 9.5: Boltzmann Machines for modeling sequence coevolution

9.2.2 *Direct-Coupling Analysis*

Principle

The starting point of Direct-Coupling Analysis is the idea, at the core of statistical physics, that short range interactions can induce long range correlations. The most famous example is the case of the Ising model: in a system of magnetic spins, short range ferromagnetic interactions, which encourage a spin to align with its neighbours can result in large scale ordering, and correlations of spin orientations spanning across scales well beyond the interaction length. In the context of protein structure predictions, we should therefore expect that interaction between residues in close vicinity may induce so-called indirect correlations between far away residues. Therefore, a good structure predictor should try to infer the underlying interaction network rather than use the correlation directly. The Direct-Coupling Analysis method, developed by Martin Weigt and collaborators [174] is one such example. It consists in modeling the MSA using a Boltzmann Machine model in which each site of the alignment is a Potts state with 20+1 state see Fig.9.5. Let $\mathbf{v}$ a sequence of the alignment. The BM defines a probability distribution over the sequence space:

$$\begin{aligned} P(\mathbf{v}) &= \frac{1}{Z} e^{-E(\mathbf{v})} \\ E(\mathbf{v}) &= -\sum_i g_i(v_i) - \sum_{i<j} J_{ij}(v_i, v_j) \end{aligned} \tag{9.2}$$

Where Z is the partition function, such that P is normalized. Here, the fields $g_i(a)$ and couplings $J_{ij}(a, b)$ model respectively conservation and coevolution and can be inferred from the data by maximum a posteriori. Besides traditional Boltzmann Machine Learning (see Section X), several more efficient inference and regularization methods were developed [82, 110, 115, 175], see [32] and [176]

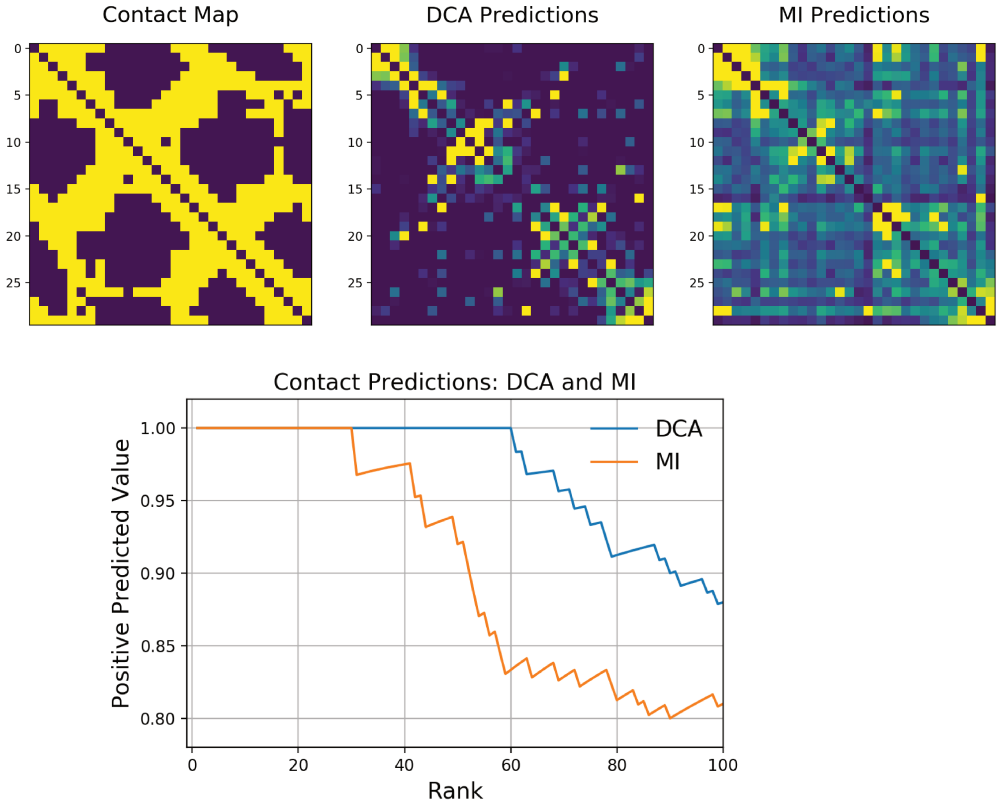

Figure 9.6: **Contact map prediction on WW using Boltzmann Machines and Mutual Information**. Top Left: True contact map, equal to 1 if residues i, j have a distance between their C_α atoms shorter than 8 Å. Top Middle: Predictions using a pairwise trained with BM learning. Top Right: Normalized mutual information between each pair of site defined as $\frac{MI(v_i,v_j)}{\min\left(S(v_i),S(v_j)\right)}$. Bottom: Fraction of correctly predicted contacts against the number of predicted contacts.

for a review. After inference, a measure of the intensity of the interactions between each pair of sites can be computed, using for instance the Frobenius norm: $J_{ij} = \sqrt{\sum_{a,b} J_{ij}(a,b)^2}$, sometimes together with the Average Product Correction [176]. In the example of the WW domain, as seen from Fig. 9.6, these interaction scores reproduce the true contact map very well. In contrast, Mutual Information produces a large number of non-zero entries (top right panel) arising from indirect correlations, resulting in larger number of false positive contacts.

Main achievements

Since their emergence, DCA and other graph-based methods such as GREMLIN [177] and PSICOV [178] have been used intensively for protein structure prediction, notably in conjunction with traditional protein folding methods. For instance, Sergey Ovchinnikov and collaborators accurately predicted two targets of the Critical Assessment for Structure Prediction 11 (CASP) by using coevolutionary contacts to constrain template-based folding within the Rosetta framework [179]. In a recent large scale study, they proposed plausible structures for representatives of 614 large protein families for which no structure was previously known [180]. Lastly, current state-of-the-art models of contact prediction from evolutionary data are supervised deep learning algorithms that combine DCA predictions with contextual information, such as solvent accessibility and secondary structure predictions [181, 182].

DCA structure prediction can also be extended to predict the structure of protein complexes. For instance, several proteins such as Hsp70 can form a complex with themselves, yielding transient contacts between sites that are distant in the tertiary structure. Since the ability to form homodimer can be important for activity regulation purpose, evolutionary pressures favoring complementary shapes can arise on these interfaces, yielding coevolution that can be detected by DCA [183]. In two systematic studies [184, 185], the authors showed that DCA could predict accurately inter-protein contacts of several known protein complexes, provided that a joint MSA of both proteins can be constructed and is large enough. In practice, this typically limits the approach to proteins families that have bacterial representatives. More recently, Yu and collaborators have shown that incorporating inter-protein coevolutionary information into docking algorithms could significantly improve prediction accuracy by sorting docking solutions found by traditional algorithms [186].

DCA can also be used for modeling the fitness landscape of proteins. It is indeed very tempting to identify the energy function inferred by DCA with the biochemical fitness of the protein. If we think of the evolutionary process as an ergodic, time-reversible exploration of a fixed fitness landscape, then there is a direct link between the equilibrum probability distribution and the underlying fitness, given by the Boltzmann law. Under this (simplistic) assumption, we can identify $E(\mathbf{v}) \equiv -\text{Fitness}(\mathbf{v})$. Recent studies have shown that the energy landscape inferred by DCA was a better fitness predictor than Position Weight Matrices (PWM), which correspond to the independent model ($J_{ij}(a, b) = 0$) [187–189]. In particular, epistatic effects, *i.e.* non-additive effects of two or more mutations could be predicted as well. Shekhar et al. [187] showed that DCA

could identify compensatory mutations in HIV, and used this information to find HIV protein sites that are least likely to mutate and escape vaccines.

Conversely, the energy function of DCA can be used to find artificial sequences with potentially high experimental fitness. In an experiment on the WW domain, Russ et al. [147] generated a library of artificial sequences by recombining the natural sequences in a way that preserves conservation and correlations - which is essentially the same thing as sampling from the Boltzmann machine probability distribution [190, 191]. They then showed that a significant fraction of these artificial sequences folded well, and had natural-like functional properties, including similar binding affinity and specificity patterns. Though DCA-generated sequences were not directly used, this experiment illustrates the potential of using a statistical energy function rather than a heuristic physical free energy function (as in folding or threading algorithms) for designing artificial sequences. Beyond structural stability, statistical energy implicitly takes into account several other factors such as binding affinity, allowing the design of putative functional proteins without knowledge of the protein-ligand complex structure, or even of the protein structure itself. From a computational efficiency perspective, we also note that Boltzmann Machines are vastly superior to threading algorithm, as they can scan very quickly through the sequence space for low energy sequences. This is done by Monte Carlo sampling of the Boltzmann distribution, at temperature $T = 1$ or $T < 1$; which directly biases the search toward low-energy sequences.

Limitations of DCA

Despite these successes, identifying the statistical energy to the experimental fitness function is too simplistic. First and foremost, the notion of fitness is plural rather than unique. Indeed, the ability for a protein to accomplish its function well in-vivo depends on numerous properties, such as:

- Stability, *i.e.* the ability for a protein to fold into a unique, low energy structure, and maintain its structure even at higher temperatures.

- Binding affinity and specificity, *i.e* the strength and specificity to which the protein binds to its target ligand. It is experimentally characterized by the catalytic rate k_{cat} and Michaelis constant K_M of the Michaelis-Menten dynamics. Within the same protein family, different proteins may have high homology and very similar structure but different binding specificities. For instance, the WW domain family can be split into four subclasses, depending on which proline-rich linear motif they recognize. Another

well known example is the case of the serine protease family: trypsin and chymotrypsin are both proteins cleavers important for digestion, but they cleave proteins at different sites.

- For enzymatic proteins, the catalytic activity, *i.e.* the efficiency at which the protein promotes a reaction of ligand. In the case of the trypsin/chymotrypsin, the reaction is the hydrolysis of the peptide bond.

- Allostery *i.e.* the ability for a protein to undergo conformational changes when binding another protein. For instance, the Hsp70 chaperone protein has two main conformations, depending on whether it is complexed to ATP or ADP. The ATP conformation allows the protein to bind to its substrate whereas the ADP conformation prevents the substrate from being released. More broadly, allostery is a fundamental mechanism for proteins involved in signaling pathways such as PDZ.

- Evolvability, *i.e.* the ability for a protein sequence to be discovered by a biological evolutionary process. For instance, a known evolutionary strategy is to reuse parts of sequences from other proteins (termed domains) for building new proteins.

Each of these properties induces a distinct evolutionary pressure, and it is the combination of all these factors that shape the probability distribution of the sequence space. In fact, these pressures can be sometimes be antagonistic: in the case of the WW domain, reaching high enough binding affinity can result in substantial loss of structural stability [192]. Therefore, although interesting similarities exist [193], it is way too simplistic to identify the statistical energy (i.e. negative log-probability) with the physical energy. In fact, several groups showed that there exist coevolutionary signals directly related to allostery [194] or binding specificity [195].

Once this point is raised, it is difficult to adapt DCA so as to disentangle the various evolutionary pressures. Expressing the energy as a sum of pairwise interactions 9.2 is well suited for finding causal links between pairs of residues, but much less effective for describing collective behaviors. In a sense, pairwise models suffer from their qualities: on the one hand, local interactions are enough for inducing large-scale collective modes, but on the other hand, characterizing and visualizing these collective modes from the fields and couplings inferred is very hard - and essentially the core of traditional statistical physics. For instance, allostery requires propagation of physical energy perturbations across many sites and as such, delicate equilibrium between each many pairs sites. Finding which of the $N(N-1)/2q^2$ statistical couplings are involved in maintaining allostery would prove very difficult. Moreover, different binding specificity

patterns can divide protein families in several subgroups, and even though Potts model can generate multimodal data (e.g. the Hopfield model), there is no simple way to recover these subgroups from the couplings.

Lastly, we should also point out that there is experimental evidence for high-order epistasis in proteins, suggesting that pairwise interactions are not enough [196, 197].

9.2.3 *Statistical Coupling Analysis and Sectors*

Principle

An interesting approach that partially addresses these issues is the Statistical Coupling Analysis and protein sectors, proposed by Ranganathan and colleagues [198, 199]. The idea is to identify subset of sites that evolve independently from one another, using a spectral analysis of the correlation matrix, similar to principal component analysis. In details, the main steps of the analysis are:

- Compute the first and second order moments $f_i(a)$, $f_{ij}(a,b)$ and covariance matrix $C_{ij}(a,b) = f_{ij}(a,b) - f_i(a)f_j(b)$
- Compute a reweighted correlation matrix $\tilde{Cij}(a,b) = C_{ij}(a,b)\phi_i(a)\phi_j(b)$, where $\phi_i^a = \log\left[\frac{f_i(a)(1-q(a))}{(1-f_i(a))q(a)}\right]$, and $q(a)$ is a baseline amino-acid distribution (e.g. $q(a) = \frac{1}{20}$ or the global empirical distribution). The purpose of the reweighting is to enhance the contribution of conserved sites in the covariance matrix.
- Sum over amino-acids $\tilde{C}_{ij} = \sqrt{\sum_{a,b} C_{ij}(a,b)^2}$, and compute the eigenvectors $\lambda_{i\mu}$ and corresponding eigenvalues.
- Select only the eigenvectors having eigenvalues significantly above the noise level. The noise level is obtained by repeating the procedure for a shuffled MSA in which all sites are independent and correlations are only due to finite sampling size. In some protein families, the top eigenvector was also discarded as well.
- For each eigenvector, identify the subgroup of sites, termed sector, having significantly large component $|\lambda_{i\mu}|$.

Sectors essentially define a partition of the sequence in which (i) intra-sector correlations are high (ii) inter-sector correlations are weak and (iii) sites not belonging of any sector have weak correlations with all sites.

Results

The sector analysis was applied to several protein families, including the WW domain [147], S1A serine protease [198], PDZ domain [200], β-lactamase [199] and Hsp70 [201]. Depending on the protein and sample size, from 1 to 3 sectors could be found and in all cases, sectors define physically contiguous regions of the protein structure. Interestingly, mutagenesis experiments suggested that different sectors control different biochemical properties of the protein. In the case of serine protease (trypsin), three sectors were found, and mutations of their respective sites impaired respectively catalytic activity, binding specificity and structural stability. In the cases of PDZ and Hsp70, one sector was found and was linked to allostery. Overall, these results suggest an organization of proteins into identifiable subgroups that are subject to distinct evolutionary pressures.

Limitations

However, sector analysis suffer from lack of statistical robustness and predictive power. Firstly, the number of relevant sectors, found through step 4 of the procedure described above, is essentially determined by the noise level of the data rather than by the protein itself; it may therefore fluctuate from one MSA to the other. Secondly, determining whether a given site belongs to a sector or not (step 5) relies on an thresholding procedure with no clear scale separation. Finally, step 2 somewhat artificially enhances the importance of conserved sites in sectors; and in cases where a single sector is extracted, it may merely consist in conserved sites rather than coevolving ones [202].

From a conceptual point of view, there is no guarantee that the various evolutionary pressures are exerted on non-overlapping subgroups of amino-acids; we rather expect them to be intertwined: for instance, a mutation important for targeting a given ligand may induce compensatory neighboring mutations to re-stabilize the structure accordingly. Lastly, the sector analysis has limited predictive power: it merely highlights sites that are more vulnerable to mutations than others, whereas DCA attempts to quantify the effect of each mutation.

10

LEARNING PROTEIN CONSTITUTIVE MOTIFS FROM SEQUENCE DATA WITH RBM

The lack of a unique, quantitative framework capable of extracting the structural and functional features common to a protein family, as well as the phylogenetic variations specific to sub-families motivates this project pursued during my PhD. Hereafter, we consider Restricted Boltzmann Machines (RBM) for this purpose. Like Boltzmann Machines / DCA, RBM is a probability distribution suited for fitness landscape predictions and sequence generation. Like Principal Component Analysis, RBM can learn a representation of the sequence space that can be related to phenotype. The difficulty lies in finding the right conditions under which RBM are good at both in the context of protein sequence modeling. Here, we show that provided a compositional regime is enforced, RBM is a powerful and versatile tool to unveil and exploit the genotype-phenotype relationship. This chapter is organized as follows. In section I, we detail the implementation of RBM in the context of protein sequence analysis. In section II, we apply RBM on several synthetic and real protein families, and show that the features inferred reflect biological properties and can be interpreted in terms of structure, function or phylogeny. In section III, we focus on structure, and present a contact map prediction algorithm based on RBM. Section IV shows sequence design applications of RBM. Section V focuses on model selection.

10.1.1 *Definition*

10.1 DEFINITION AND IMPLEMENTATION

In the context of protein sequence analysis, a Restricted Boltzmann Machine (RBM) is a joint probabilistic model for sequences and representations, see Fig. 10.1. Protein sequences $\mathbf{v} = (v_1, v_2, ..., v_N)$ are displayed on the Visible layer, and representations $\mathbf{h} = (h_1, h_2, ..., h_M)$ on the Hidden layer. Each visible unit takes one out of $q = 21$ values (20 amino acids + 1 alignment gap). Depending on the potential, hidden-layer unit values h_μ are either real or binary. The formal definition is very similar to that of the binary case:

Joint probability distribution The joint probability distribution of $\mathbf{v}, \mathbf{h}$ is:

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{N} g_i(v_i) - \sum_{\mu=1}^{M} \mathcal{U}_\mu(h_\mu) + \sum_{i,\mu} h_\mu \, w_{i\mu}(v_i) \right), \tag{10.1}$$

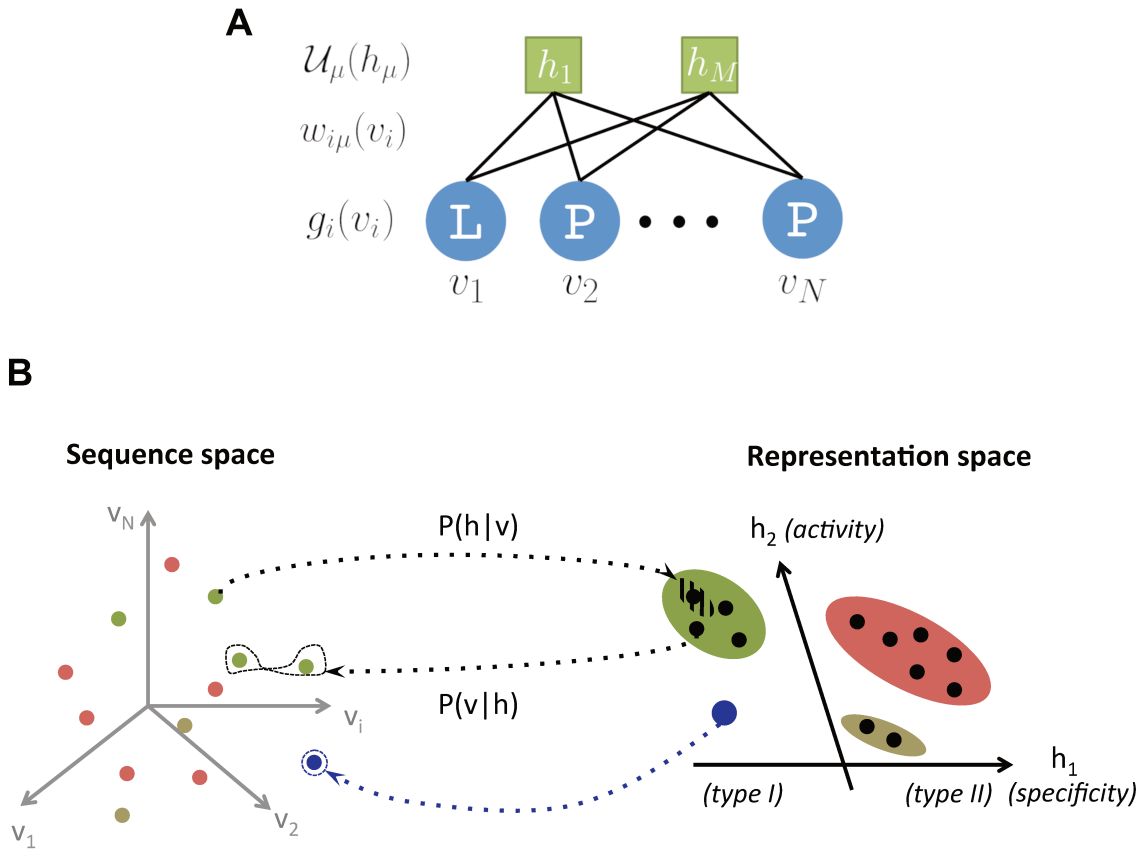

Figure 10.1: **Reverse and forward modeling of proteins. A.** A Restricted Boltzmann Machine (RBM) for protein sequence modeling. Weights $w_{i\mu}$ connect the visible layer (carrying protein sequences $\mathbf{v}$) to the hidden layer (carrying representations $\mathbf{h}$). Biases on the visible and hidden units are introduced by the local potentials $g_i(v_i)$ and $\mathcal{U}_\mu(h_\mu)$. **B.** Sequences $\mathbf{v}$ in the MSA (dots in sequence space, left) code for proteins with different phenotypes (dot colors). RBM define a probabilistic mapping from sequences $\mathbf{v}$ onto the representation space $\mathbf{h}$ (right), indicative of the phenotype of the corresponding protein, and encoded in the conditional distribution $P(\mathbf{h}|\mathbf{v})$ (black arrow). The reverse mapping from representations to sequences is $P(\mathbf{v}|\mathbf{h})$ (black arrow). Sampling a subspace in the representation space (colored domains) defines in turn a complex subset of the sequence space (colored circles), and allows one to design new sequences with putative phenotypic properties (blue domain and arrow).

Where Z is the usual partition function. Here, the fields $\mathbf{g}$ and weights $\mathbf{w}$ are tensors of size respectively $N \times q$ and $M \times N \times q$, indexed by the visible layer index i, amino-acid index v and for $\mathbf{w}$ the hidden layer index μ.

Hidden unit input Given a sequence $\mathbf{v}$ on the visible layer, hidden unit μ receives the following input I_μ:

$$I_\mu(\mathbf{v}) = \sum_i w_{i\mu}(v_i) \ . \tag{10.2}$$

This expression is analogous to the score of a sequence with a position-specific weight matrix. Large and small $|I_\mu|$ correspond to, respectively, good and bad matches between the weights and the sequence.

Hidden unit potential As for binary visible units, the input I_μ determines the conditional probability of the activity h_μ of the hidden unit:

$$P(h_\mu|\mathbf{v}) \propto \exp\big(-\mathcal{U}_\mu(h_\mu) + h_\mu \, I_\mu(\mathbf{v})\big) \ , \tag{10.3}$$

up to a normalization constant. The nature of the potential $\mathcal{U}$ is crucial to determine how the average activity $\langle h_\mu|\mathbf{v}\rangle$ varies with the input I. Unless stated explicitely, we use a dReLU potential for $\mathcal{U}_\mu$, see Section 3.2. For dReLU potentials, the average activity is an adaptive non-linear function of the input, that can interpolate between linear, ReLU, sigmoid, and double ReLU. dReLU potentials can adjust to gaussian, sparse, multimodal or skewed input distributions and induces effective high-order interactions in the visible layer. We justify the choice of dReLU potential over Bernoulli and quadratic potentials in Section V.

From representation to sequence Given a representation (set of activities) $\mathbf{h}$ on the hidden layer, the residues on site i are distributed according to

$$P(v_i|\mathbf{h}) \propto \exp\left(g_i(v_i) + \sum_\mu h_\mu \, w_{i\mu}(v_i)\right) \ . \tag{10.4}$$

Hidden units with large activities h_μ strongly bias this probability, and favor values of v_i corresponding to large weights $w_{i\mu}(v_i)$.

Gauge choice Since the conditional probability Eqn. 10.4 is normalized, the transformations $g_i(a) \to g_i(a) + \lambda_i$ and $w_{i\mu}(a) \to w_{i\mu}(a) + K_{i\mu}$ leave the conditional probability invariant. We choose the zero-sum gauges, defined by $\sum_v g_i(v) = 0$, $\sum_v w_{i\mu}(v) = 0$.

Marginal probability distribution The probability of a sequence, $P(\mathbf{v})$, is obtained by summing (integrating) $P(\mathbf{v}, \mathbf{h})$ over all its possible representations $\mathbf{h}$:

$$P(\mathbf{v}) = \int \prod_{\mu=1}^{M} dh_\mu P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{N} g_i(v_i) + \sum_{\mu=1}^{M} \Gamma_\mu\big(I_\mu(\mathbf{v})\big) \right) \tag{10.5}$$

Where $\Gamma_\mu(I) = \log\left[\int dh\, e^{-U_\mu(h) + h\, I} \right]$ is the cumulant generative function associated to the potential U_μ. Its derivative with respect to the input, $\frac{\partial \Gamma_\mu}{\partial I}$, is the average activity of hidden unit μ

Sampling As for binary RBM, sampling from $P(v, h)$ is obtained by alternating sampling from $P(h|v)$ and $P(v|h)$. We discuss in Section IV biased sampling techniques relevant for protein design.

10.1.2 *Learning*

As for binary data, the weights $w_{i\mu}(v)$ and the defining parameters of the potentials $g_i(v)$ and $\mathcal{U}_\mu$ are learned by maximizing the average likelihood $\langle \log P(\mathbf{v}) \rangle_{MSA}$ of all sequences $\mathbf{v}$ within the Multiple Sequence Alignment (MSA). To correct for the heterogeneous sampling of the sequence space (some animal kingdoms such as primates are over-represented against others such as archaea), we apply a standard reweighting scheme: each sequence $\mathbf{v}^\ell$ with $\ell = 1, ..., B$ is assigned a weight w_ℓ equal to the inverse of the number of sequences with more than 90% amino-acid identity (including itself). In all that follows, the average over the sequence data of a function f is defined as

$$\langle f(\mathbf{v}) \rangle_{MSA} = \left(\sum_{\ell=1}^{B} w_\ell\, f(\mathbf{v}^\ell) \right) \Big/ \left(\sum_{\ell=1}^{B} w_\ell \right) . \tag{10.6}$$

We also add penalty terms over the weights and fields to prevent overfitting and to create interpretable sequence representations. A standard L_2 regularization term over the fields $\propto g_i(v)^2$ prevents them from diverging when an amino-acid was never seen at a given position. A L_1^2 regularization term over the weights $\propto \sum_\mu \left(\sum_{i,v} |w_{i\mu}(v)| \right)^2$ is crucial to avoid overfitting and to produce sparse weights. As shown in part iii, sparse weights are crucial for learning

compositional representations. Though sparsity naturally emerge during training in MNIST, it does not in proteins, and sparsity must be enforced. Overall, the cost function writes:

$$\langle \log P(\mathbf{v}) \rangle_{MSA} - \frac{\lambda_f}{2} \sum_{i,v} g_i(v)^2 - \frac{\lambda_1^2}{2qN} \sum_{\mu} \left(\sum_{i,v} |w_{i\mu}(v)| \right)^2 , \tag{10.7}$$

Choosing the value of the sparse penalty is not trivial but not arbitrary; we discuss rationales for this choice in Section V. As for binary data, the optimization is carried out by stochastic gradient ascent, evaluating the model averages by Monte Carlo. We initialize the fields with the ones of the best fitting independent model:

$$g_i^0(v) = \log \langle \delta_{v_i,v} \rangle_{MSA} - \frac{1}{q} \sum_{v} \log \langle \delta_{v_i,v} \rangle_{MSA} \tag{10.8}$$

And the weights and hidden potentials are initialized as usual. During training, the only notable difference is that the gauges must be maintained. For the fields $g_i(a)$, the gradient updates directly preserve the zero-sum gauge. For the weights, we add the following line after each update:

$$w_{i\mu}(v) = w_{i\mu}(v) - \frac{1}{q} \sum_{v'} w_{i\mu}(v') \tag{10.9}$$

Lastly, we have used traditional Persistent Contrastive Divergence for the training algorithm, with from 1 to 10 Monte Carlo steps. Tests on small proteins showed that provided the model is regularized, there is little improvement in likelihood between traditional sampling and Parallel Tempering / Augmented Parallel Tempering methods. We did not have time to investigate this on larger proteins.

10.1.3 *Weight Visualization*

To visualize the weights tensors inferred by the machine, we introduce the weight logo representation. Each weight logo represents a weight attached to one hidden unit μ, $w_{i\mu}(v)$. As in a sequence logo, the x-axis is the site index. At each site, the height of each letter is proportional to the corresponding weight coefficient $w_{i\mu}(v)$; positive weights are above the x-axis and negative weights below; letters are sorted by weight amplitude. Examples follow in next section.

11

A PHENOMENOLOGY OF FEATURES INFERRED BY RBM

We present in this section results of RBM trained on five protein families:

(a) Lattice-protein *in silico* data [203, 204] to benchmark our approach on an exactly solvable model with known fitness function [205].

(b) The WW domain, a short module binding different classes of ligands [206] important for signaling pathway.

(c) The Kunitz domain, a protease inhibitor, historically important for protein structure determination [207].

(d) The Serine protease protein family, an important family of protein-cleaving enzymes such as trypsin.

(e) The Hsp70 protein, a large chaperone protein [208]

We have found structure-related features, either local, such as tertiary contacts, or extended, such as secondary structure motifs (α-helix and β-sheet) or characteristic of intrinsically disordered regions (2) functional features, i.e. groups of amino acids controlling specificity or activity; (3) phylogenetic features, related to sub-families sharing evolutionary determinants. Some of these features involves two residues only (as direct pairwise couplings do), others extend over large and not necessarily contiguous portions of the sequence (as in collective modes extracted with PCA). A selection of features follows now. We kindly warn the physicist reader that this section might be hard for the eye, as we will dive head-first into the terminology-rich world of structural biology. The features found will be extensively compared to current knowledge about these proteins, acquired through years of genomics, structural studies and mutagenesis experiments. The main purpose of this section is to show that RBM can extract very detailed and specific information about each protein family. But beyond this zoology of proteins and features, we hope to convince the reader that RBM open a window into the general principles underlying natural protein design.

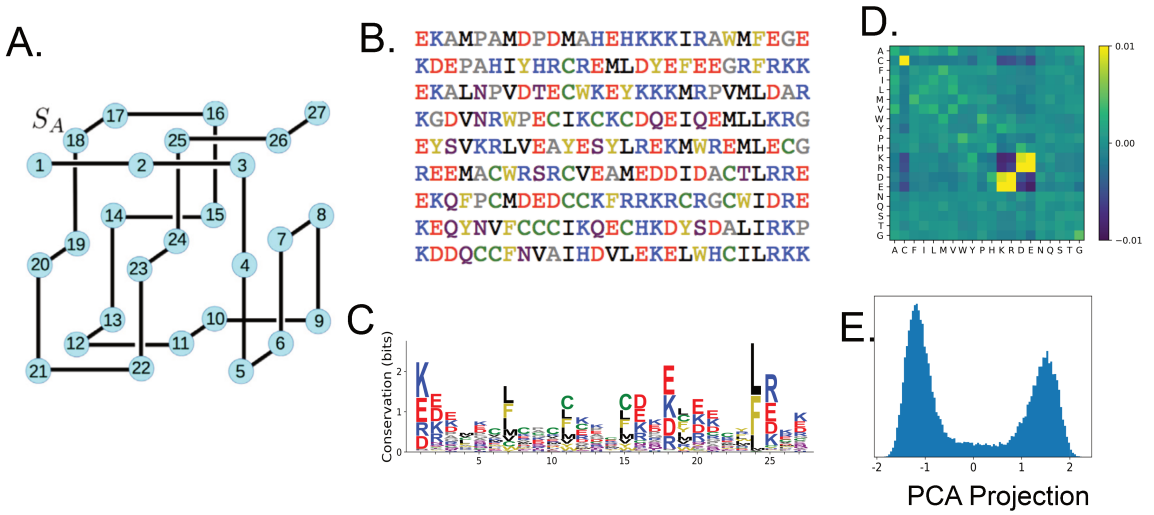

Figure 11.1: **Lattice Protein (LP) Sequence alignments** LP are synthetic multiple sequence alignments that share several features with natural multiple sequence alignments. (a) A native structure S_A (b) A subset of sequences generated by Monte Carlo that fold specifically in S_A (c) Sequence logo of the MSA, featuring non-conserved and partly conserved sites. (d) Average covariance between pairs of sites in contact: $C(a,b) = \frac{1}{28}\sum_{<i,j>} C_{ij}(a,b)$, highlighting the preferred amino-acid interaction. (e) Histogram of the first principal component of the MSA: sequence cluster into two subfamilies.

11.1 LATTICE PROTEINS

11.1.1 *Description*

Lattice Proteins models have been introduced in the '90 to investigate the uniqueness of folding shared by the majority of real proteins [203, 204]. In this model, a 'structure' is defined as a self-avoiding path of the 27 amino-acid-long chains, on a $3 \times 3 \times 3$ lattice cube [203]. Each structure defines a set of 28 tertiary contacts. A 'protein' of $N = 27$ amino acids may fold into one of the $\mathcal{N} = 103,406$ possible structures, with probabilities depending on its sequence. The probability that the protein sequence $\mathbf{v} = (v_1, v_2, ..., v_{27})$ folds in one of these, say, S, is

$$P_{nat}(\mathbf{v};S) = \frac{e^{-\mathcal{E}(\mathbf{v};S)}}{\sum_{S'=1}^{\mathcal{N}} e^{-\mathcal{E}(\mathbf{v};S')}} , \tag{11.1}$$

where the energy of sequence $\mathbf{v}$ in structure S is given by

$$\mathcal{E}(\mathbf{v};S) = \sum_{i<j} c_{ij}^{(S)}\, E(v_i, v_j) . \tag{11.2}$$

In the formula above, $c^{(S)}$ is the contact map: $c_{ij}^{(S)} = 1$ if the pair of sites ij is in contact, *i.e.* i and j are nearest neighbors on the lattice and zero otherwise. The pairwise energy $E(v_i, v_j)$ represents the amino-acid physico-chemical interactions, given by the the Miyazawa-Jernigan (MJ) knowledge-based potential [209].

A MSA of 36,000 sequences that fold specifically on structure S_A, *i.e.* with high probability $P_{nat}(\mathbf{v}; S_A) > 0.99$, through Monte Carlo sampling from the Boltzmann distribution with $\mathcal{H} \propto -\log P_{nat}$ [205]. As in real MSA, Lattice Protein data feature conservation, short- and long-range correlations between amino-acid on different sites, as well as high-order interactions that arise from competition between folds [205], see Fig. 11.1. However, unlike real proteins, the fitness function - structural stability - is mathematically well defined and it is also fairly intuitive: a good protein must fold specifically into its native conformation; otherwise it is useless half of the time. Moreover, sequences are statistically independent and the MSA can be arbitrarily large, so noise levels are arbitrarily low. LP are therefore great candidates for benchmarking RBM.

11.1.2 *Results*

A RBM with $M = 100$ dReLU hidden units and $\lambda_1^2 = 0.025$ is learned from the MSA. We present in Fig. 11.2 a selection of structural LP features inferred by the model. For each hidden unit μ, we show in panel A the weight logo of $w_{i\mu}(v)$ and in panel B the distribution of its hidden unit input I_μ, as well as the conditional mean function $\langle h_\mu | I_\mu \rangle$. In all cases, the weights are significant only for a limited number of sites; this will guide our interpretation. As seen from panel A, weight 1 focuses mostly on sites 3 and 26, which are in contact in the structure (black contour). Positively charged residue (H,R,K) have a large positive (resp. negative) component on site 3 (resp. 26), and negatively charged residues (E,D) have a large negative (resp. positive) components on the same sites. The histogram of its input distribution (panel B) shows three main peaks in the data. Since $I_1(v) = \sum_i w_{i1}(v_i)$, the peaks (i) $I_1 \sim 3$, (ii) $I_1 \sim -3$ and (iii) $I_1 \sim 0$ correspond respectively to sequences having (i) positively charged amino-acids at site 3 and negatively charged amino-acids at site 26 (ii) conversely, negatively charged amino-acids at site 3 and positively charged at site 26 and (iii) identical charges or non-charged amino-acids. We knew a priori that this pair of sites could take different values since they are not very conserved (see sequence logo Fig. 11.1 D), but the fact that hidden unit 1 focuses on these sites signals an *excess* of sequences having significantly high $|I_1|$ compared to an independent model. Indeed, the contribution of the hidden

unit to the log-probability is $\Gamma_1(I_1) \sim I_1^2$, since the conditional mean Γ_1' is close to linear (see panel B). In other words, sequences folding into S_A often have residues with opposite charges on sites 3 and 26, forming an electrostatic contact (or salt bridge). Feature 2 is another weight focusing on sites 3 and 26. Its positive components are similar to the negative components of weight 1, but the negative components corresponds to hydrophobic amino-acids (I,L,V,M,A) in both sites 3 and 26. The negative peak at $I_2 \sim -2$ therefore identifies sequences having hydrophobic amino-acids at both sites. To summarize, here, 'evolution' favored sequences having complementary amino-acids at sites 3 and 26, and the resulting statistical signal (positive and negative correlations) was caught by hidden units 1 and 2.

Interestingly, RBM can extract features involving more than two sites. Weight 3 and 4 are related to, respectively, the triplets of neighboring amino acids 8-15-27 and 2-16-25, each realizing two overlapping contacts on S_A (blue and orange dashed contours). Both highlight collective mode spanning over more than two sites: sequences having very negative $I_3 \sim -2$ are characterized by an electrostatic 'triangle' $(15,+) \leftrightarrow (8,-) \leftrightarrow (27,+)$, whereas sequences having very negative $I_4 \sim -2$ have all three sites 2-16-25 occupied by hydrophobic amino-acids. Both subsets are relatively small but again, it is still in excess with respect to what would be expected from an independent model. In fact, the strong non-linearities could even suggest an excess with respect to a pairwise model. This will be discussed later in Section V.

Weight 5 is located mainly on sites 5 and 22, with weaker weights on sites 6, 9,11. It codes for a cisteine-cisteine disulfide bridge located on the bottom of the structure and present in about a third of the sequences ($I_5 \sim 3$). The weak components and small peaks $I_5 \sim 4$ also highlight sequences with a triplet of cisteines. We note however that this is an artifact of Lattice Proteins, as a cisteine may form only one disulfide bridge.

Weight 6 is an extreme version of the electrostatic triangle. It has important components on sites 23,2,25,16,18 corresponding to the upper side of the protein. Again, the region is contiguous, and the weight logo indicates a pattern of alternate charges present in many sequences ($I_6 \gg 0$ and $I_6 \ll 0$).

The collective modes defined by RBM may not be contiguous. Weight 7 codes for an electrostatic triangle 20-1-18, and the electrostatic 3-26, which is far away from the former. This indicates that despite being far away, sites 1 and 26 often have the same charge. The latter constraint is not due to the native but impedes folding in the 'competing' structure, S_G, in which sites 1 and 26 are neighbours. Such so-called negative design was also reported through analysis with pairwise model [205]. Besides weight 7, we have found other weights

indicating negative design; in particular pairs of distant sites that must not have cisteine together, as they would form a disulfide bridge in structure S_G.

To summarize, we have shown that RBM can retrieve many biological features directly related to the underlying structure and competitors of the protein. The scenario was of course ideal owe to small protein size, simple fitness function and infinite sampling. We note however that many initial implementations failed to pass this test. We now turn to real-protein families.

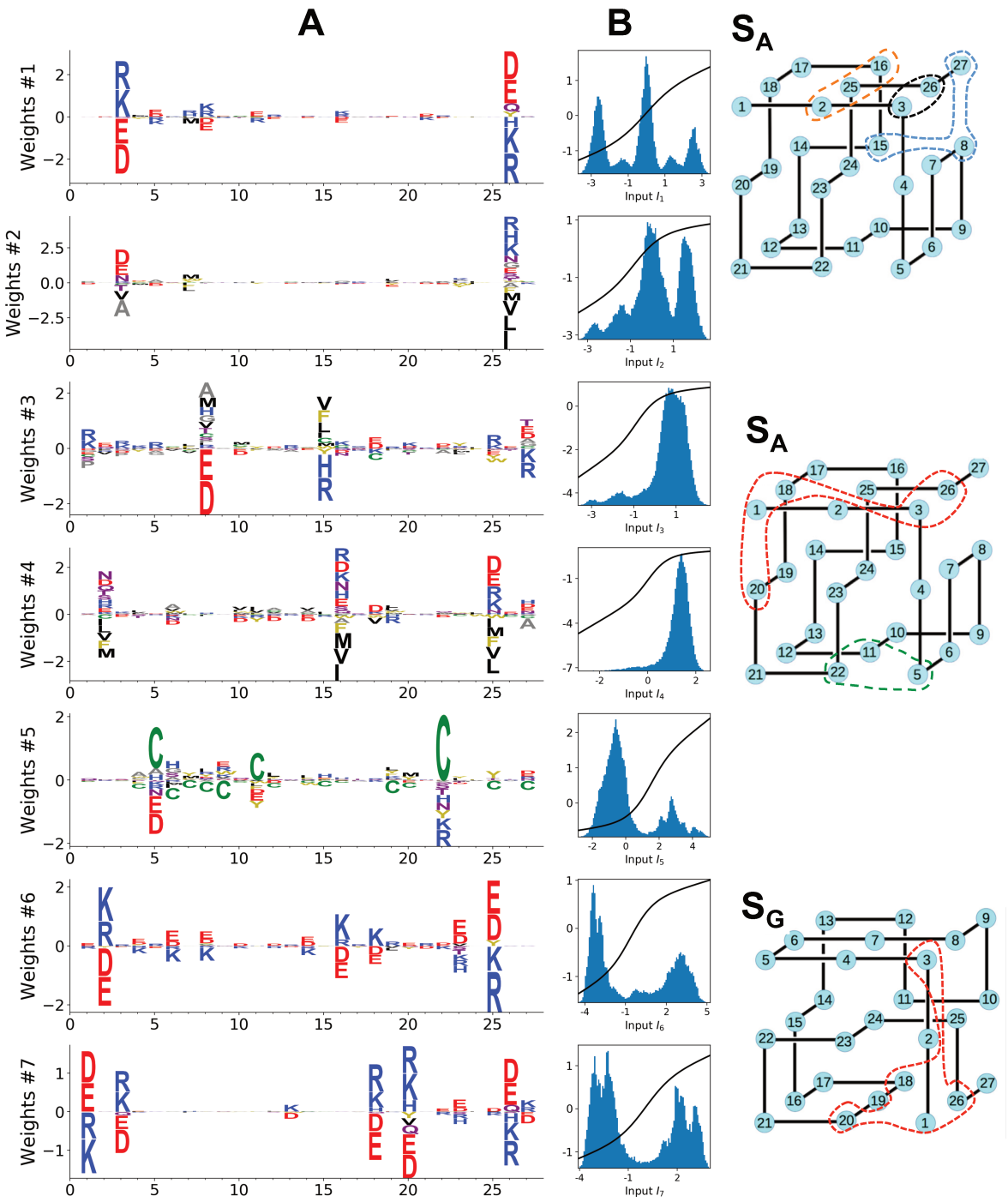

Figure 11.2: **Modeling Lattice Proteins with RBM A.** Seven weight logos, visualizing the weights $w_{i\mu}(v)$ attached to seven selected hidden units. **B** Distribution of inputs received by the corresponding hidden units, and conditional mean function (full line and left scale). Right: Structure of S_A and S_G, one of S_A's main competing structure with dashed contour highlighting the main sites of hidden units 1/2 (black), 3 (blue), 4 (orange), 5 (green) and 7 (red).

11.2 WW DOMAIN

11.2.1 *Description*

The majority of natural proteins are obtained by concatenating functional building blocks, called protein domains, that can fold and function independently of the rest of the sequence. The WW domain is one of the smallest domains, with $N = 31$ residues. WW is a protein-protein interaction domain found in many eukaryotes and human signalling proteins, involved in essential cellular processes such as transcription, RNA processing, protein trafficking, receptor signalling. One example of WW-domain is YAP1, a protein that activates the transcription of genes involved in cell proliferation and suppresses apoptotic genes. It folds into a three-stranded antiparallel β-sheet, see Fig. 11.4D. The domain name stems from the two conserved tryptophans (W) at positions 5-28 (Fig. 3A), which serve as anchoring sites for the ligands. WW domains bind to a variety of proline (P)-rich peptide ligands, and can be divided into four groups, based on their preferential binding affinity [210]. Group I binds specifically to PPXY motif - where X is any amino acid; Group II to PPLP motifs; Group III to proline-arginine containing sequences (PR); Group IV to phosphorylated serine/threonine-proline sites [p(S/T)P]. Modulation of binding properties allow hundreds of WW domain to specifically interact with hundreds of putative ligands in mammalian proteomes.

11.2.2 *Results*

We have trained a RBM with dReLU potential,$M = 100$ hidden units and $\lambda_1^2 = 0.25$ on the PFAM alignment PF00397. We show five hidden units, with their weight logos and corresponding input distribution in Fig.11.4 B,C. We also map the important sites of each weight logo onto the structure of WW in Fig. 11.4D. In all following, a site i is considered important if $\sum_v |w_{i\mu}(v)| > 40\% \times \max_i \sum_v |w_{i\mu}(v)|$. Lastly, we show the distribution of Hamming distances (*i.e.* fraction of sites with different residues) within the alignment, and within the top-20 sequences that have highest activation on the feature. This allows us to check whether the feature inferred is activated only for a small subset of the sequence space or for many distantly related sequences.

Weight 1 is reminiscent of Lattice Proteins, as it codes for a contact between sites 4-22 realized either by two amino acids with oppositive charges ($I_1 < 0$), or by one tiny and one negatively charged amino acid ($I_1 > 0$).

Weight 2 shows a β-sheet–related feature, with large entries defining a set of mostly hydrophobic ($I_2 > 0$) or hydrophilic ($I_2 < 0$) residues localized on the β_1 and β_2 strands and in contact on the 3D fold. The input distribution, with a large peak on negative I_2, suggest that this part of the WW domain is in contact with the solvent in most, but not all, natural sequences.

Hidden unit 3 is negatively activated by few evolutionary-related sequences (see Hamming distance distribution) carrying the W28X mutation, with non-aromatic X; this rare mutation is accompanied by a complex mutation pattern around the β_1-β_2 extremities. Notably, many sequences with positive I_3 have a glycine at site 14, whereas those with negative I_3 do not have it, having either a glycine or a gap at site 15. Since glycine often appear right before β strands, this suggests a slightly different structure compared to consensus. This is consistent with the observation that sequences lacking tryptophan are not functional for linear motif recognition, suggesting a completely different functional role.

Weights 4 and 5 involve sites on the β_2-β_3 binding pocket and on the β_1-β_2 loop of the WW domain. The distributions of activities highlight different groups of sequences in the MSA that strongly correlate with experimental ligand-type identification, see Fig. 11.3. We find that (i) Type I domains are characterized by $I_4 < 0$ and $I_5 > 0$; (ii) Type II/III domains are characterized by $I_4 > 0$ and $I_5 > 0$; (iii) There is no clear distinction between Type II and Type III domains; (iv) Type IV domains are characterized by $I_4 > 0$ and $I_5 < 0$. These findings are in good agreement with various studies:

(i) Mutagenesis experiment have shown the importance of sites 19, 21, 24, 26 for binding specificity [211, 212]. For the YAP1 WW domain, as confirmed by various studies (see [212] Table 2), the mutations H21X and T26X reduce the binding affinity to Type I ligands, while Q24R increases it and S12X has no effect. This is in agreement with the negative components of weight 4 : I_4 increases upon mutations H21X and T26X, decreases upon Q24R and is unaffected by S12X. Moreover the mutation L19W alone, or combined with H21[D/G/K/R/S] could switch the specificity from Type I to Type II/III [211]. These results are consistent with Fig. 11.3 YAP1 (blue cross) is of Type I but one or two mutations move it to the right side, closer to the other cluster (orange crosses). Espanel and Sudol [211] also proposed that Type II/III specicity required the presence of an aromatic amino acid (W/F/Y) on site 19, in good agreement with weight 3.

(ii) The distinction between Types II and III is unclear in the literature, because WW domains often have high affinity with both ligand types.

(iii) Several studies [147, 192, 213] have demonstrated the importance of the β_1-β_2 loop for achieving Type IV specificity, which requires a longer, more

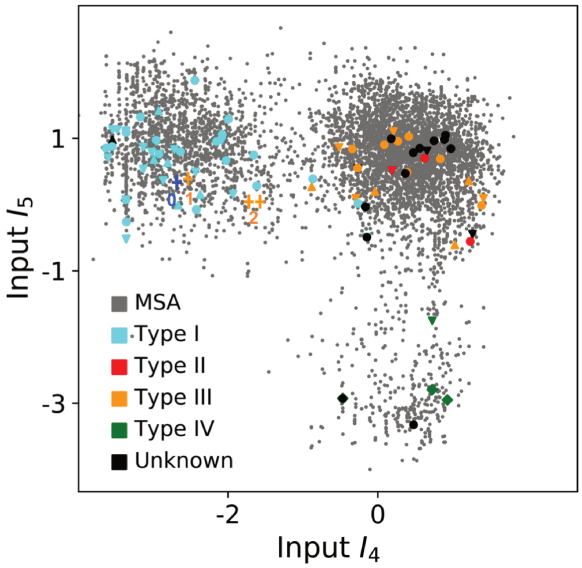

Figure 11.3: Scatter plot of inputs I_4 vs. I_5. Gray dots represent the sequences in the MSA; they cluster into three main groups. Colored dots show artificial or natural sequences whose specificities, given in the legend, were tested experimentally. Upper triangle: natural, from [147]. Lower triangle: artificial, from [147]. Diamond: natural, from [214]. Crosses: YAP1 (0) and variants (1 and 2 mutations from YAP1), from [211]. The three clusters match the standard ligand type classification.

flexible loop, as opposed to short rigid loop for other types. The length of the loop is encoded in weight 4 through the gap symbol on site 13: short and long loops correspond to, respectively, positive and negative I_5. The importance of residues R11 and R13 was shown in [213] and [147], where removing R13 of Type IV hPin1 WW domain reduces its binding affinity to [p(S/T)P] ligands. These observations agree with weight 4, authorizing substitutions between K and R on sites 11 and 13.

(iv) A specificity-related sector of eight sites was identified in [147], five of which carry the top components of weight 4 (green balls in Fig. 11.4D). Our approach not only provides another specificity-related feature (weight 5) but also the motifs of amino acids affecting Type I & IV specificity, in good agreement with the experimental findings of [147].

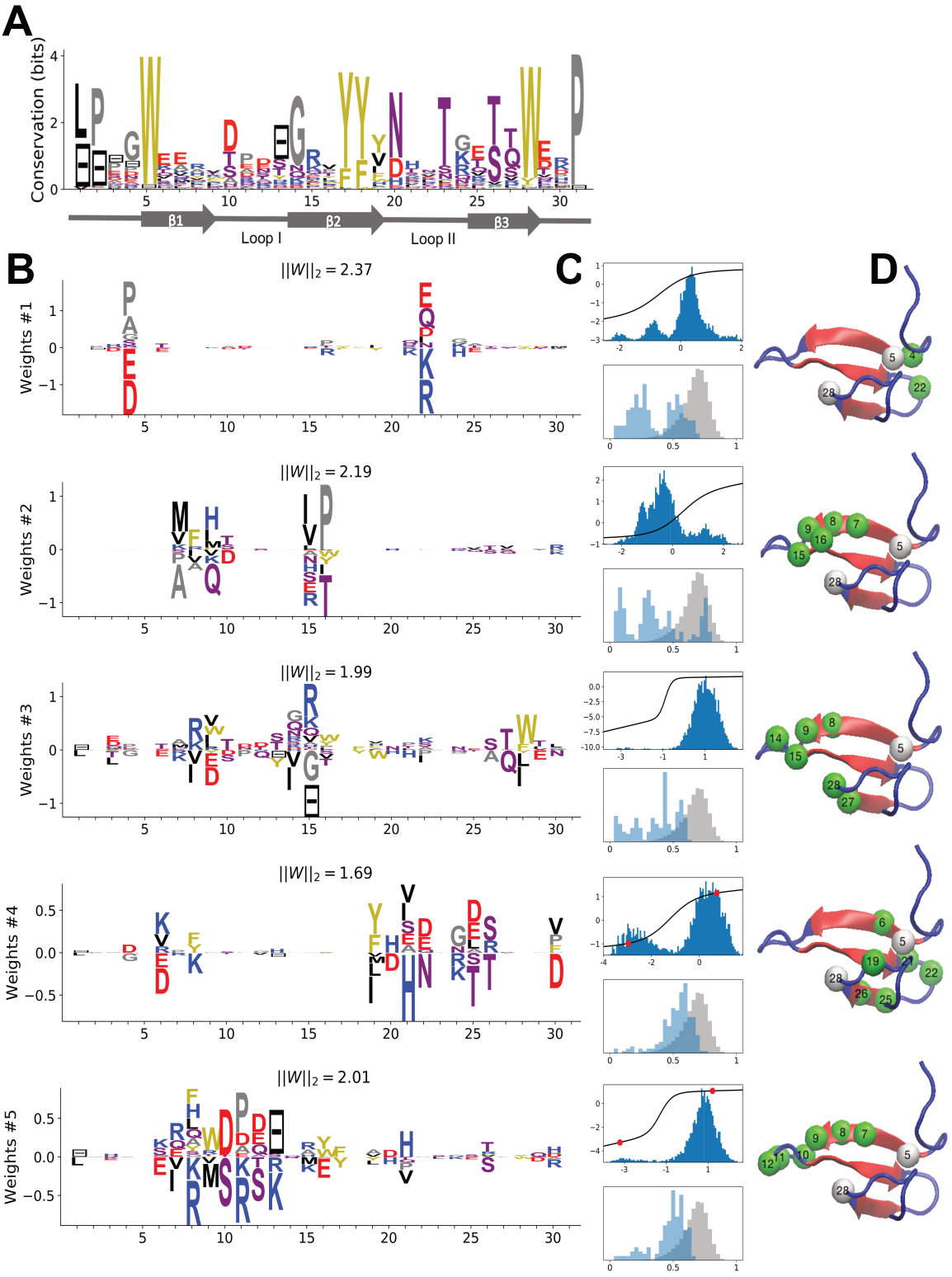

Figure 11.4: **Modeling WW Domain with RBM. A.** Sequence logo and secondary structure of the WW domain (PF00397) **B.** Weight logos for four representative hidden units **C.** Corresponding inputs, average activities and distances between top-20 feature activating sequences **D.** 3D visualization of the features, shown on the PDB structure 1eom [146]. White spheres locate the two W. Green spheres locate residues i with high $\sum_v |w_{i\mu}(v)|$

11.3 KUNITZ DOMAIN

11.3.1 *Description*

The Kunitz domain, with $N = 53$ residues is present in several genes and its main function is to inhibit serine protease such as trypsin. Kunitz domains play a key role in the regulation of many important processes in the body such as tissue growth and remodeling, inflammation, body coagulation and fibrinolysis. They are implicated in several diseases such as tumor growth, Alzheimer, cardiovascular and inflammatory diseases and therefore they have been largely studied and shown to have a large potential in drug-design [215, 216].

Some examples of Kunitz domain-containing proteins include the Basic Pancreatic Trypsin Inhibitor (BPTI, 1 Kunitz domain), the Bikunin (2 domains) [217], Hepatocyte growth factor activator inhibitor (HAI, 2 domains) and tissue factor pathway inhibitor (TFPI, 3 domains) [215, 216].

Structurally, the Kunitz domain is characterized by 2 $\alpha-$ helices and 2 $\beta-$ strands and, as frequently observed for small protein, cysteine-cysteine disulfide bridges largely contribute to its thermodynamic stability. Figure 11.5A shows the MSA sequence logo and the secondary structure motifs. BPTI structure was the first one ever resolved [207], and is often used to benchmark folding predictions based on simulations [218] and coevolutionary approaches [175, 177, 219–221].

11.3.2 *Results*

We have trained a RBM with dReLU potential,$M = 100$ hidden units and $\lambda_1^2 = 0.25$ on the PFAM alignment (PF00014 $B = 7503$ sequences, [222])

Weight 1 in has large components on sites 45 and 49, in contact in the final α_2 helix. The distribution of the inputs I_1 partitions the MSA in three subfamilies (top histogram). The two peaks in $I_1 \simeq -2.5$ and $I_1 \simeq 1.5$ identify sequences in which the contact is due to an electrostatic interaction with, respectively, $(+,-)$ and $(-,+)$ charged amino acid on sites 45 and 49; the other peak in $I_1 \simeq 0$ identify sequences realizing the contact differently, e.g. with an aromatic amino acid on site 45. Weight 1 shows also a weaker electrostatic component on site 53; the 4-site separation between sites 45–49–53 fits well with the average helix turn of 3.6 amino acids.

Weight 2 focuses on the contact between residues 11-35, realized in most sequences by a C-C disulfide bridge (negative I_2 peak in input distribution). A

minority of sequences in the MSA, corresponding to $I_2 > 0$ and mostly coming from nematode organisms (Fig. 11.6A), do not show the bridge. A subset of these sequences strongly and positively activate hidden unit 3 ($I_3 > 0$ peak in input distribution and Fig. 11.6A). Positive components in weight 3 logo suggest that these proteins stabilize their structure through electrostatic interactions between sites 10 ($-$ charge) and 33-36 ($+$ charges both, see Fig. 11.6B), to compensate the absence of C-C bridge on the neighbouring sites 11-35.

Both weights 4 and 5 describe features mostly localized on the loop preceding the β_1-β_2 strands (sites 7 to 16). Structural studies of the trypsin-trypsin inhibitor complex have shown that this loop binds to the proteases [223]; site 12 is notably in contact with the active site of the protease and is therefore key to the inhibitory activity of the domain. The two amino acids (R, K) having a large positive contribution to weight 4 in position 12 are basic and bind to negatively charged residues (D, E) on the active site of trypsin-like serine protease. While several Kunitz domains with known trypsin inhibitory activity, such as BPTI, TFPI, TPPI-2,... give rise large and positive inputs I_4, Kunitz domains with no trypsin/chymotrypsin inhibition activity, e.g. associated to COL7A1 and COL6A3 genes [224, 225], correspond to negative or vanishing values of I_4. Hence, hidden unit 4 possibly separates the Kunitz domains having trypsin-like protease inhibitory activity from the others.

This interpretation is also in agreement with mutagenesis experiments carried out on sites 7 to 16 to test the inhibitory effects of Kunitz domains BPT1, HAI-1, and TFP1 against trypsine-like proteases [215, 216, 226–228]. In [226] it was shown that mutation R12A on the first domain (out of two) of HAI-1 destroyed its inhibitory activity; a similar effect was observed in the presence of non basic residues on site 12 in the first two domains (out of three) of TFP1 as discussed in [216]. The affinity between human serine proteases and the mutants G9F, G9S, G9P of bovine BPTI was shown to decrease in [227]. Conversely, in [225] it was shown that the set of mutations P10R, D13A, F14R could convert the COL6A3 domain into a Trypsin inhibitor. All these results are in agreement with the above interpretation of weight 4. Note that, though quite few sequences have large I_4, many correspond to small or negative values. This may be explained by the facts that (i) many of the Kunitz domains analyzed are present in two or more copies, and as such, are not all required to strongly bind trypsin [216] and (ii) Kunitz domain may have other specificities encoded by other hidden units. In particular, weight 5 displays on site 12 large components associated to medium to large size hydrophobic residues (L, M,Y), and is possibly related to other serine protease specificity classes such as chymotrypsin.

Weight 6 is an example of phylogenetic feature. It codes for a complex extended mode, negatively activated by a small subset of the MSA composed

of evolutionary close sequences (see Hamming distance distribution). These sequences correspond to the protein Bikunin present in most mammals and some other vertebrates [215]. In our analysis, most protein families exhibited several phylogenetic modes with distribution similar to weight 6.

Lastly, we show in Fig.11.7 a selection of so-called gap modes. Gap modes code for long stretches of gaps, often but not always located at the extremeties of the sequence [220]. They are activated by the few sequences within the alignment that lack the corresponding missing sites. These sequences are often, but not always evolutionary close (see panel D). Though gap modes are essentially artifacts of the alignment procedure, visual inspection suggests that their distribution of positions may not be random. In some cases, it seems that they extend exactly over secondary structure elements of the protein, such as β strands. We have found gap modes in every real protein family studied, and further investigation of this effect would be very interesting.

11.4 TRYPSIN AND SERINE PROTEASE

11.4.1 *Description*

Serine protease are enzymes that cleave peptide bonds in proteins. They are found in both eukaryotes and prokaryotes, and involved in various physiological process such as digestion and blood coagulation and immune response. Some examples include trypsin and chymotrypsin, whose role is to cleave nutrient proteins for digestion, elastase, which break down membrane proteins of bacteria, and plasmin, which degrades blood plasma proteins. Structurally, serine protease have length about 220 residues, and are composed by two beta-barrels converging at the catalytic site, as well as 4 alpha-helices and six disulfide bridges, see Fig. 11.9. All members of the family share a common catalytic mechanism for cleaving proteins, involving a catalytic triad of Histidine, Serine, and Threonine [230]. A catalysis event begins by the insertion of the peptide bond within the catalytic triad, followed by a binding of the hydroxyl group of the Serine to the carbonyl group of the peptide bond (hence the name), and results in hydrolysis of the bond. Depending on the composition of the active site [231, 232], Serine-Protease target specific peptide bonds: Trypsin specifically cleave bonds containing a positively charged amino-acid (R,K), whereas Chymotrypsin targets hydrophobic aromatic amino-acids (F,Y,W) and elastase targets small amino-acids (A,G,V).

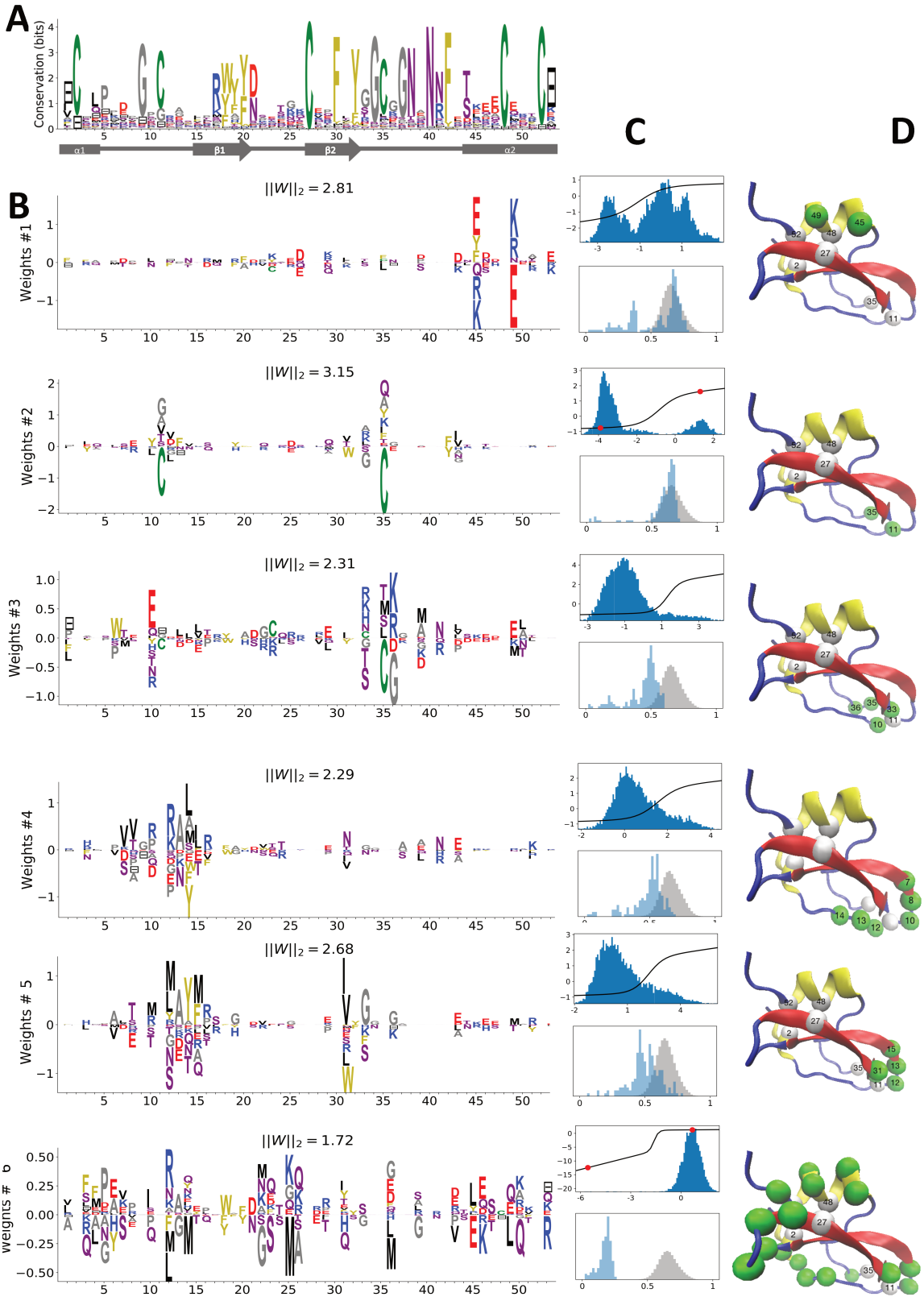

Figure 11.5: **Modeling Kunitz Domain with RBM.** Same panels as Fig. 11.4. The weights are visualized on PDB structure 2knt [229]. White spheres denote the positions of the 3 disulfide bridges in the wild type sequence.

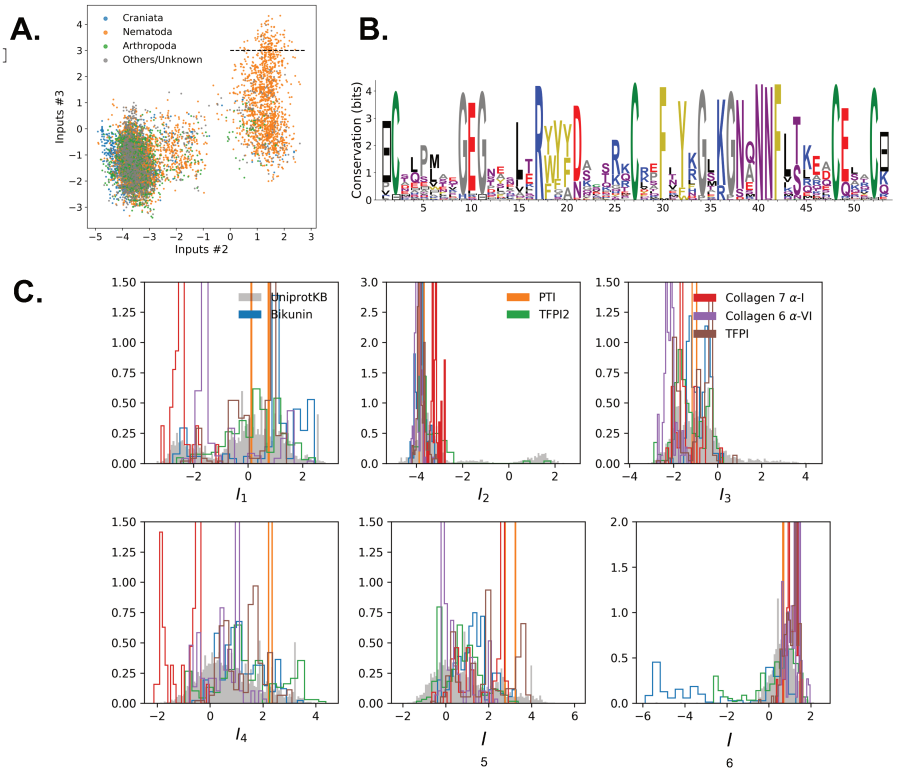

Figure 11.6: **Identifying phylogenetic identity of feature-activating sequences A.** Scatter plot of inputs of hidden units 2 and 3. Most of the sequences that lack the disulfide bridge come from nematodes **B** Sequence logo of the 137 sequences above the dashed line ($I_3 > 3$), showing the electrostatic triangle that putatively replaces the disulfide bridge. **C** Input distribution for 5 Kunitz-containing genes

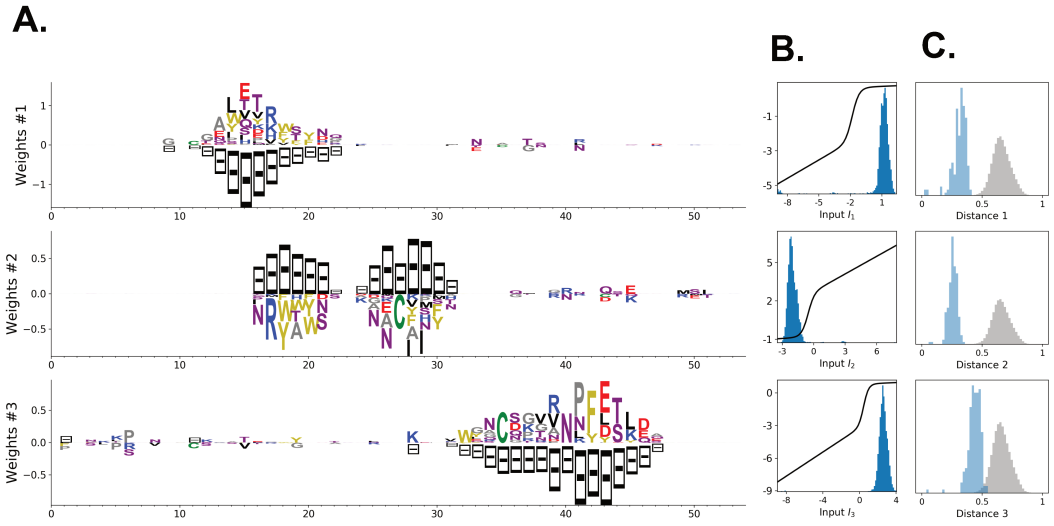

Figure 11.7: **Example of gap modes for the Kunitz domain** Gap modes arise in almost all protein families, and model sequences lacking several sites of the protein.

The mechanism of action and specificity of this family are well understood, with a large body of computational studies such as sector analysis [198, 199]. One particular topic of interest is to better understand how did the serine protease family evolve to diversify its functionalities [233], and in particular how can one given protein evolve into a different functionality [234, 235]. We present in the next section features differentiating the various subfamilies, and discuss in chapter IV how RBM can probe these functional transitions.

11.4.2 *Results*

We have trained a RBM with $M = 200$ dReLU hidden units and $\lambda_1^2 = 0.25$, on the MSA from [189], with $B = 47913$ sequences and $N = 217$ sites. We found, as usual several traditional gap modes and structural features, such as disulfide bridge and contact modes. Moreover, we found several hidden units with bimodal input distribution reminiscent of what was found in the WW domain. To assess whether these modes separate the various subfamilies described above, we used available labeled data from UniprotKB [199], and looked for subgroups of hidden units that partitioned the sequence space into functionally distinct regions. This is done automatically as follows:

- Each hidden unit defines a binary of the sequence space, e.g. with $\Gamma'_\mu(I_\mu) \lessgtr 0$. The minimum of Γ corresponds to a minimum of probability for the hidden input I_μ; it matches in practice a gap between two modes when the distribution is bimodal.
- Each set of l hidden units defines a partition of the sequence space into 2^l subsets. There are $\begin{pmatrix} M \\ l \end{pmatrix}$ such partitions.
- For fixed $l = 2, 3, 4$, we looked for the partitions that maximize the mutual information between the partition index $\text{Part}(\mathbf{v}) \in [1, 2^l]$ and the functional class.

Here, sequences were split in 7 functional subgroups: Trypsin, Chymotrypsin, Tryptase, Kallikrein, Granzyme, Elastase, Haptoglobin. We chose $l = 4$ in this example, and selected the first four features shown in Fig. 11.8 among the best partitions.

As seen from the input histograms and scatter plots in Fig.11.9, the combination of hidden units allows to separate well the different subclasses. In particular, weight 1 separates sequences with trypsin specificity from sequences

with chymotrypsin specificity. It is experimentally known that D168 and S168 are respectively important for Trypsin and Chymotrypsin specificity, in agreement with weight 1. The other sites are located around the active site of the enzyme, in agreement with their presumed functional role, see Fig. 11.9C. Similarly, weight 5 is localized on the catalytic triad, and separates the haptoglobin (which are non-enzymatic) from the rest of sequences.

Lastly, weight 6 codes for a collective mode located on the surface of the protein. Its amino-acid content is very similar for all positions, with negative components for uncharged hydrophilic residues (A,G,S,T,N,Q) and positive components for charged residues (K,R,H,E,D). Inspection of the sequences having large positive or negative I_6 shows that some sequences have as few as one charged residue over the 59 most important important sites of weight 6, whereas other sequences have more than 30. Weight 6 therefore separates proteins based on their surface charge density. Functionally, this could be related to the modulation of the enzyme's activity by pH. Another possible explanation is related to autolysis, namely the ability for one protease to cleave another protease. Autolysis, in particular for trypsin, is key for the regulation of the protease's concentrations, and higher charge density may lead to higher electrostatic repulsion and thus reduced autolysis [236]. Though further investigation is clearly required, we remark that this is an interesting example of biological feature with a graded input distribution rather than bimodal; it illustrate the importancy of using flexible hidden unit potentials rather than simply binary.

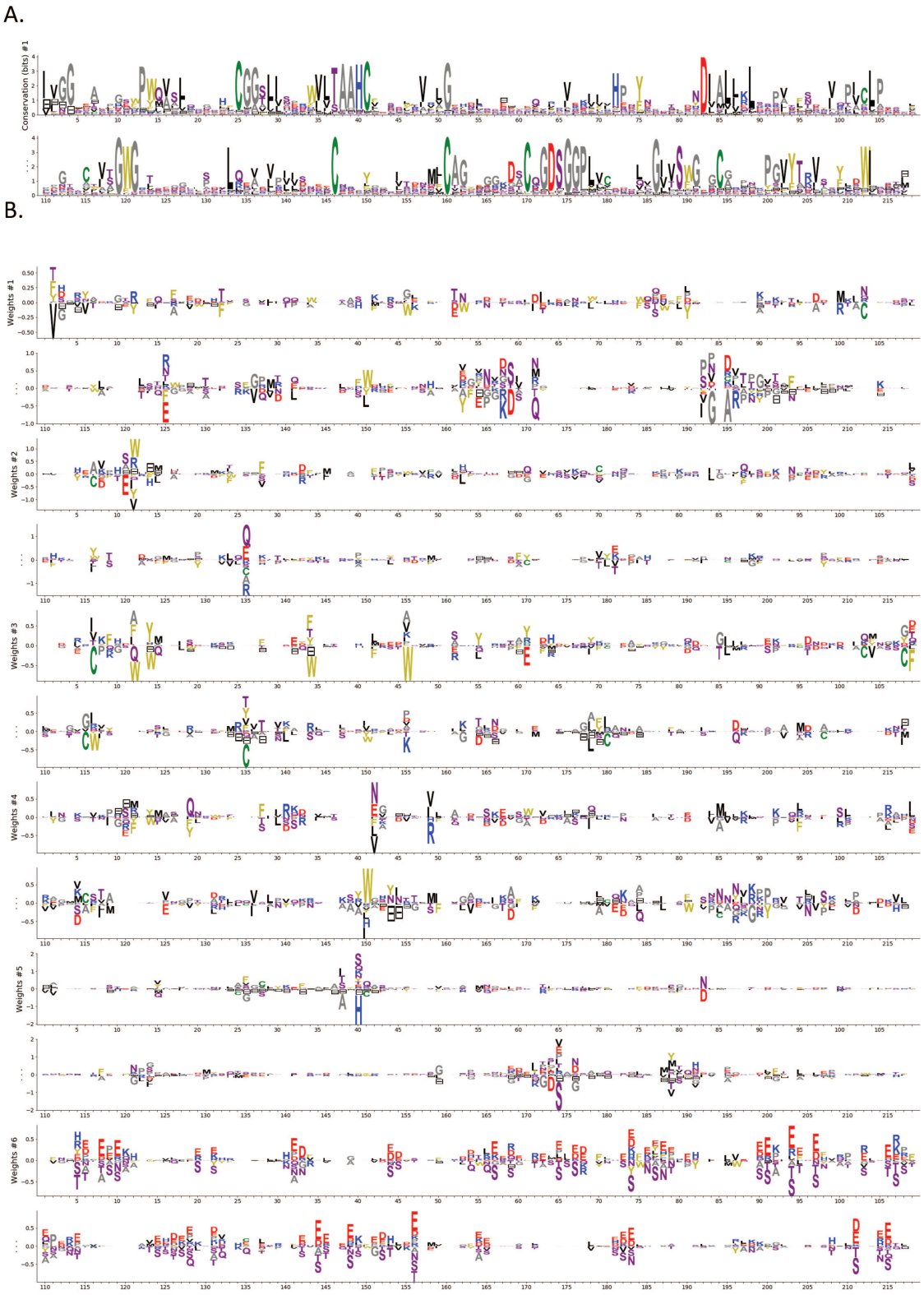

Figure 11.8: **Modeling Serine Protease with RBM.** A. Sequence logo and B. Selected weight logos

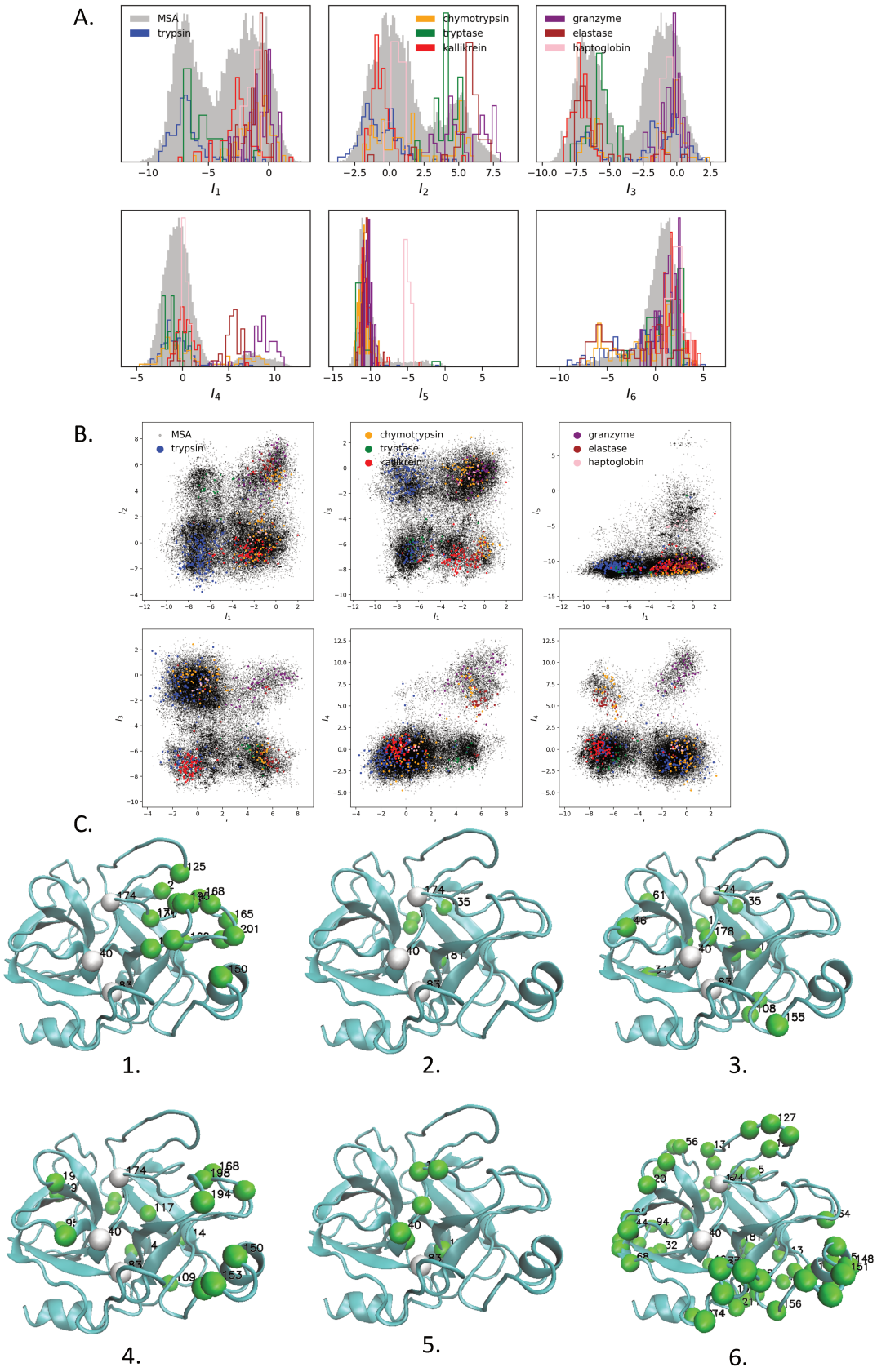

Figure 11.9: **Modeling Serine Protease with RBM.** Corresponding A. Input distribution, B. input scatter plots and C. Structures, on PDB 3TGI [237]. White spheres denote the catalytic triad

11.5 HSP70 PROTEIN

11.5.1 *Description*

70-kDa heat shock proteins (Hsp70) form a highly-conserved family represented in essentially all organisms. Hsp70, together with other chaperone proteins, perform a variety of essential functions in the cell: they can assist folding and assembly of newly synthetized proteins, trigger refolding cycles of misfolded proteins, transport unfolded proteins through organelle membranes, and when necessary, deliver non-functional proteins to the proteasome, endosome or lysosome for recycling [208, 238, 239]. There are 13 HSP70s protein-encoding genes in humans, differing by where (nucleus/cytoplasm, mitochondria, endoplasmic reticulum) and when they are expressed. Some, such as HSPA8 (Hsc70) are constitutively expressed whereas others such as HSPA1 and HSPA5 are stress-induced (respectively by heat shock and glucose deprivation). Notably, Hsc70 can make up to 3% of the total total mass of proteins within the cell, and is thus one of its most important housekeeping genes. Structurally, Hsp70 are multi-domain proteins of length of 600-670 sites (631 for E-Coli DNaK gene). They consist of

- A Nucleotide Binding Domain (NBD, 400 sites) that can bind and hydrolyse ATP. It is homologous to other ATPase domains such as the one in Actin [240].
- A Substrate Binding Domain (SBD sites), folded in a beta-sandwich structure, which binds to the target peptide or protein.
- A flexible, hydrophobic interdomain-linker linking the NBD and the SBD.
- A LID domain, constituted by several (up to 5) α helices, which encapsulates the target protein and blocks its release.
- An unstructured C-terminal tail of variable length, important for detection and interaction with other co-chaperones, such as Hop proteins [241].

Hsp70 functions by adopting two different conformations, see Figs. 11.11C&D. When the NBD is bound to ATP, the NBD and the SBD are held together and the LID is open, such that the protein has low binding affinity to substrate peptides. After hydrolysis of ATP to ADP, the NBD and the SBD detach from one another, and the LID is closed, yielding high binding affinity and effectively trapping the peptides between the SBD and the LID. By cycling between both conformations, Hsp70 can bind to misfolded proteins, unfold them by stretching (e.g. with two

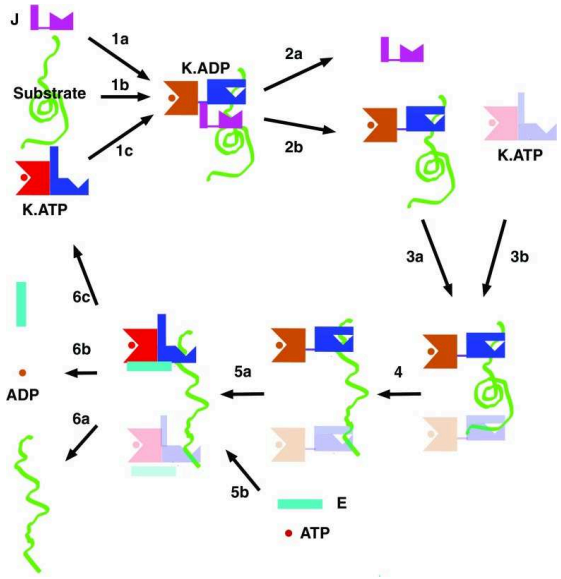

Figure 11.10: **Hsp70 functional cycle** Graphical summary reproduced from [239]. Red/Brown = Nucleotide Binding Domain. Blue = Substrate Binding Domain. Green = Substrate protein. Purple = J-protein. Cyan = Nucleotide Exchange Factor

Hsp70 bound at two ends of the protein) and release them for refold cycles. Since Hsp70 alone have low ATPase activity, this cycle requires another type of co-chaperone, J-protein, which simultaneously binds to the target protein and the Hsp70 to stimulate its ATPase activity, as well as a Nucleotide Exchange Factor (NEF) that favors swaps of the ADP back to ATP and hence release of the target protein. Fig. 11.10 summarizes the Hsp70 functional cycle; it is reproduced from the review by Zuiderweg et al. [239].

11.5.2 *Results*

We have constructed a multiple sequence alignment for HSP70 with $N = 675$ residues and $B = 32,170$ sequences: starting from the seeds of [183], and querried SwissProt and Trembl UniprotKB databases using HMMER3 [171]. Annotated sequences were grouped based on their phylogenetic origin and functional role. Prokaryotes mainly express two Hsp70 proteins: DnaK ($B = 17,118$ sequences in the alignment), which are the prototype Hsp70, and HscA ($B = 3,897$), which are specialized in chaperoning of Iron-Sulfur cluster containing proteins. Eukaryotes Hsp70 were grouped by location of expression (Mitochondria: $B = 851$, Chloroplaste: $B = 416$, ER: $B = 433$, Nucleus/Cytoplasm and others: $B = 1,452$). We also singled out Hsp110

sequences, which, despite the high homology with Hsp70, correspond to non-allosteric proteins ($B = 294$). We have then trained a dReLU RBM over the full MSA with $M = 200$ hidden units. We show below the weight logos and input distributions for ten selected hidden units.

Nucleotide Binding Domain Weights 1,2,3 focus on the loop within the IIB subdomain of the NBD, see Fig. 11.11A,B. As seen from the stretches of gaps, both weights 1 and 2 encode for a variability of the length of the loop. Depending on the sequence, the loop can be long ($I_1, I_2 > 0$), short ($I_1 < 0$,$I_2 > 0$ 4-5 sites shorter) or very short ($I_1, I_2 < 0$ 8-10 sites shorter). This classification corresponds respectively to the Prokaryotic DnaK, Eukaryotic Hsp70 and Prokaryotic HscA. This structural difference between the three families was previously reported and is of high functional importance to the NBD [242, 243]. Shorter loops increase the nucleotide exchange rates (and thus the release of target protein) in the absence of NEF, and the loop size controls interactions with NEF proteins [243–245]. Hsp70 proteins having long and intermediate loop size interact specifically with respectively GrpE and Bag-1 NEF proteins, whereas short, HscA-like loops did not interact with any of them. This cochaperone specificity allows for functional diversification within the cell; for instance, Eukaryotic Hsp70 proteins expressed within mitochondria and chloroplasta, such as the human gene HSPA9 and the Chlamydomonas reinhardtii HSP70B share the long loop with prokaryotic DNaK proteins, and therefore do not interact with Bag proteins. As shown by weight 3, the amino-acid content of the loop also varies within the prokaryotic DNaK subfamily, with at least two distinct subfamilies (middle and right peaks of I_3). Though we did not find mention of these subfamilies in the literature, they suggest a diversity of NEF-protein specificity within the DNaK subfamiliy.

Feature-based classification of Eukaryotic Hsp70 Weight 4 encodes a small collective mode localized on $\beta_4 - \beta_5$ strands, at the edge of the β sandwich within the SBD. Weight are quite large ($w \sim 2$), and the input distribution is bimodal, separating notably HscA and chloroplastal Hsp70 ($I_2 > 0$) from mitochondrial Hsp70 and the other Eukaryotic Hsp70 ($I_2 < 0$). We note also a similarity in structural location and amino-acid content with weight 4 of the WW–domain, which controls binding specificity (Fig. 11.3). We have not found trace of this motif in the literature either, but its location, strength and amino-acid content suggest that it could be important for binding substrate specificity. Besides chloroplastic and mitochondrial-specific Hsp70, we also found an inter–domain mode separating Endoplasmic reticulum-specific Hsp70 proteins from the other Eukaryotic proteins (Weight 5, green spheres in Fig. 11.12, weight logo not shown).

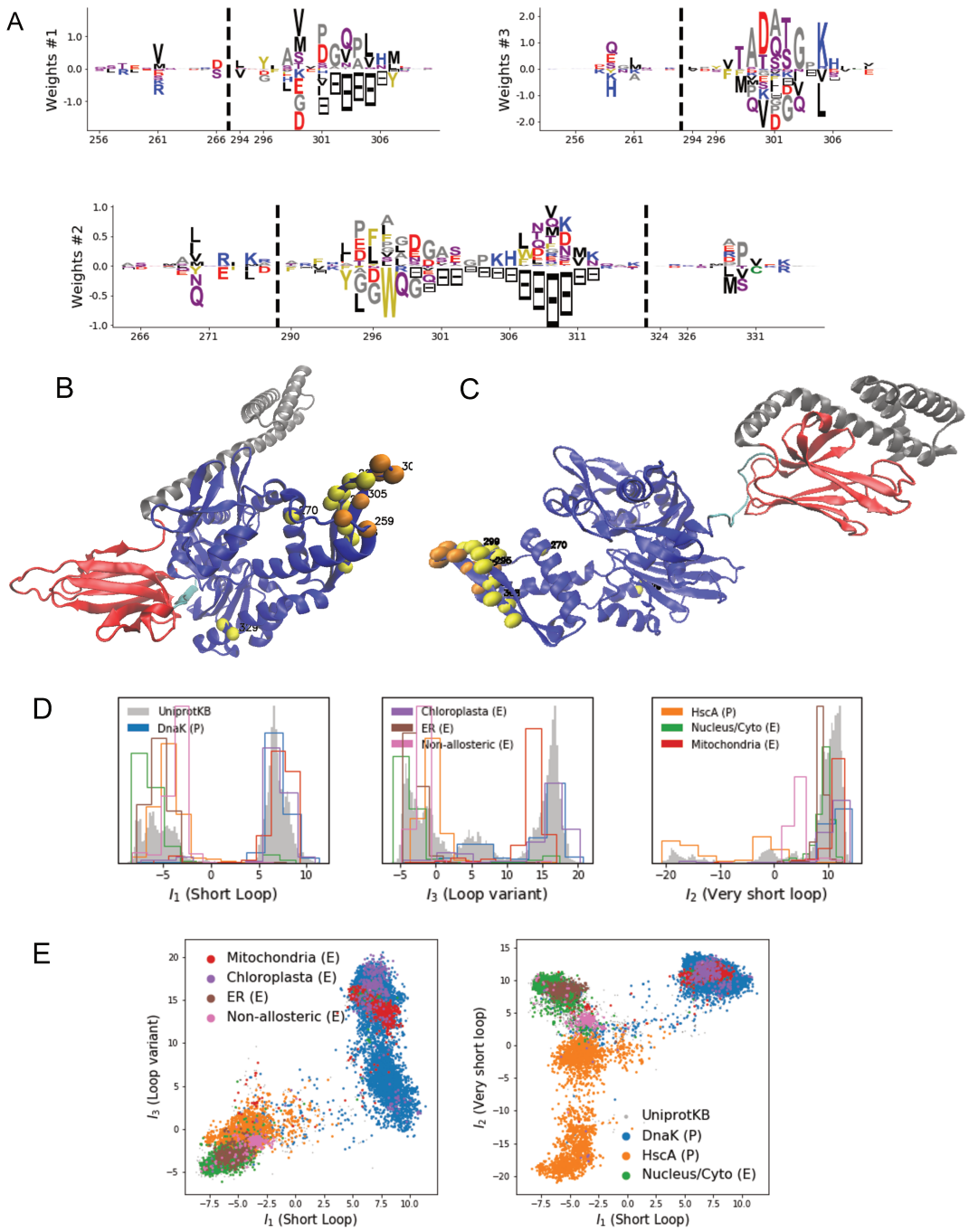

Figure 11.11: **Modeling HSP70 with RBM: Nucleotide Binding Site A.** Truncated sequence logo of weights 1-3. Due to the large protein length, we show only weights for positions i with large weights ($\sum_v |w_{i\mu}(v)| > 0.4 \times \max_i \sum_v |w_{i\mu}(v)|$), with surrounding positions up to ± 5 sites away; dashed lines vertical locate the left edges of the intervals. **B.** Structure of the ATP-bound state conformation of E-Coli DNaK (PDB: 4jne [246]), and **C.** of the ADP-bound state (PDB: 2kho [247]).Protein backbone colors: Blue=NBD, Cyan=Linker, Red=SBD, Gray=LID. Orange spheres = weights 1 and 3; Yellow spheres = weight 2 **C.** Corresponding input distribution, and **D.** Scatter plots

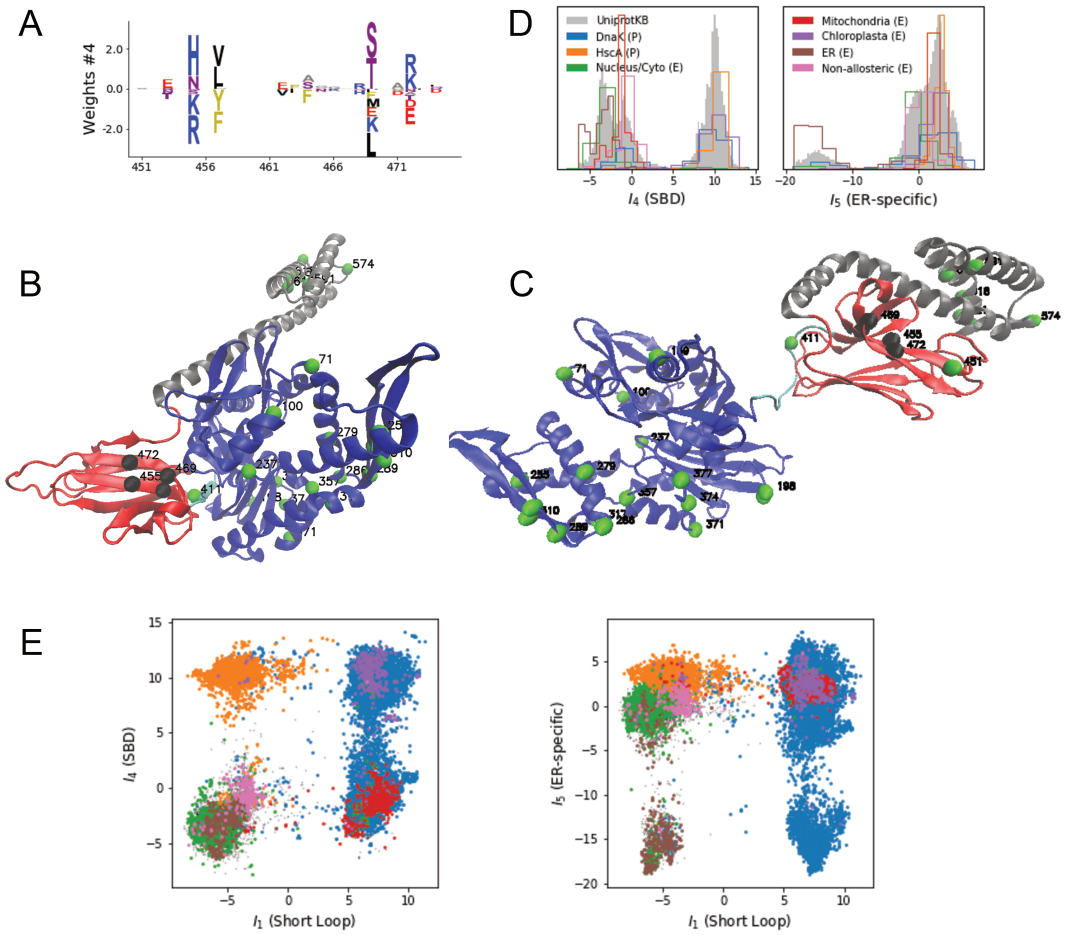

Figure 11.12: **Modeling HSP70 with RBM: Classification A.** Truncated sequence logo of weight 4 **B.** Structure of the ATP-bound state conformation of E-Coli DNaK (PDB: 4jne [246]), and **C.** of the ADP-bound state (PDB: 2kho [247]). Same backbone colors as above. Black spheres = weights 4; Green spheres = weight 5 **D.** Corresponding input distribution, and **E.** Scatter plots

Inter-domain collective modes and allostery RBM can also extract collective modes of coevolution spanning multiple domains, such as weights 6,7,8. The residues supporting Weight 6 are physically contiguous in the ADP conformation, but not in the ATP conformation, see Fig. 11.13 (weight logo not shown). Hence, weight 6 captures inter-domain coevolution between the SBD and the LID domains.

Weight 7 also codes for a wide, inter–domain collective mode, localized at the interface between the SBD and the NBD domains. When the Hsp70 protein is in the ATP conformation, the sites carrying weight 7 are physically contiguous, whereas in the ADP state they are far apart. Moreover, its input distribution separates the non-allosteric Hsp110 subfamily ($I_4 \sim 0$) from the other subfamilies ($I_4 \sim 40$), suggesting that this motif is important for allostery. Weight 8 is another weight separating non-allosteric from allosteric sequences. Several mutational studies have highlighted 21 important sites for allostery within E-Coli DNaK [201]; 7 of these positions are present in the top 38 most important sites of Weight 7, 4 appear in Weight 8, and several others are highly conserved and do not coevolve at all.

Unstructured tail Weight 9 codes for a collective mode located mainly on the unstructured C-terminal tail, with few sites on the LID domain, see Fig. 11.14A. [1]. Its amino-acid content is strikingly similar across all sites: positive weights for hydrophilic residues (in particular, lysine), and negative weights for tiny, hydrophobic residues. Indeed, as seen from Fig. 11.14B-D hydrophobic-rich or hydrophilic-rich sequences are found in the MSA. This motif is consistent with the role of the tail for cochaperone interaction: hydrophobic residues are important for formation of Hsp70-Hsp110 complexes via the Hop protein [241]. High-charge content is also frequently encountered and at the basis of recognition mechanism in intrinsically disordered protein regions [248], which could suggest the existence of different protein partners.

Dimerization In its ATP-bound conformation, the Hsp70 protein can form an antiparallel homo-dimer. This dimer is formed with the help of J-protein, and presumed to facilitate transfer of the substrate protein to another chaperone protein, Hsp90 [249, 250]. We found a statistical trace of this homo-dimer: Weight 10 codes for a collective mode located on two sides of the protein, that are in contact in the dimer, see Fig. 11.15. Its input distribution is trimodal, which could suggest different dimerization modalities, or some subgroups that do not form any dimer at all.

Comparison with other methods Some of the results presented here were previously obtained with others coevolutionary methods. In [183], the authors

1 The structure is not shown since the tail cannot be crystallized

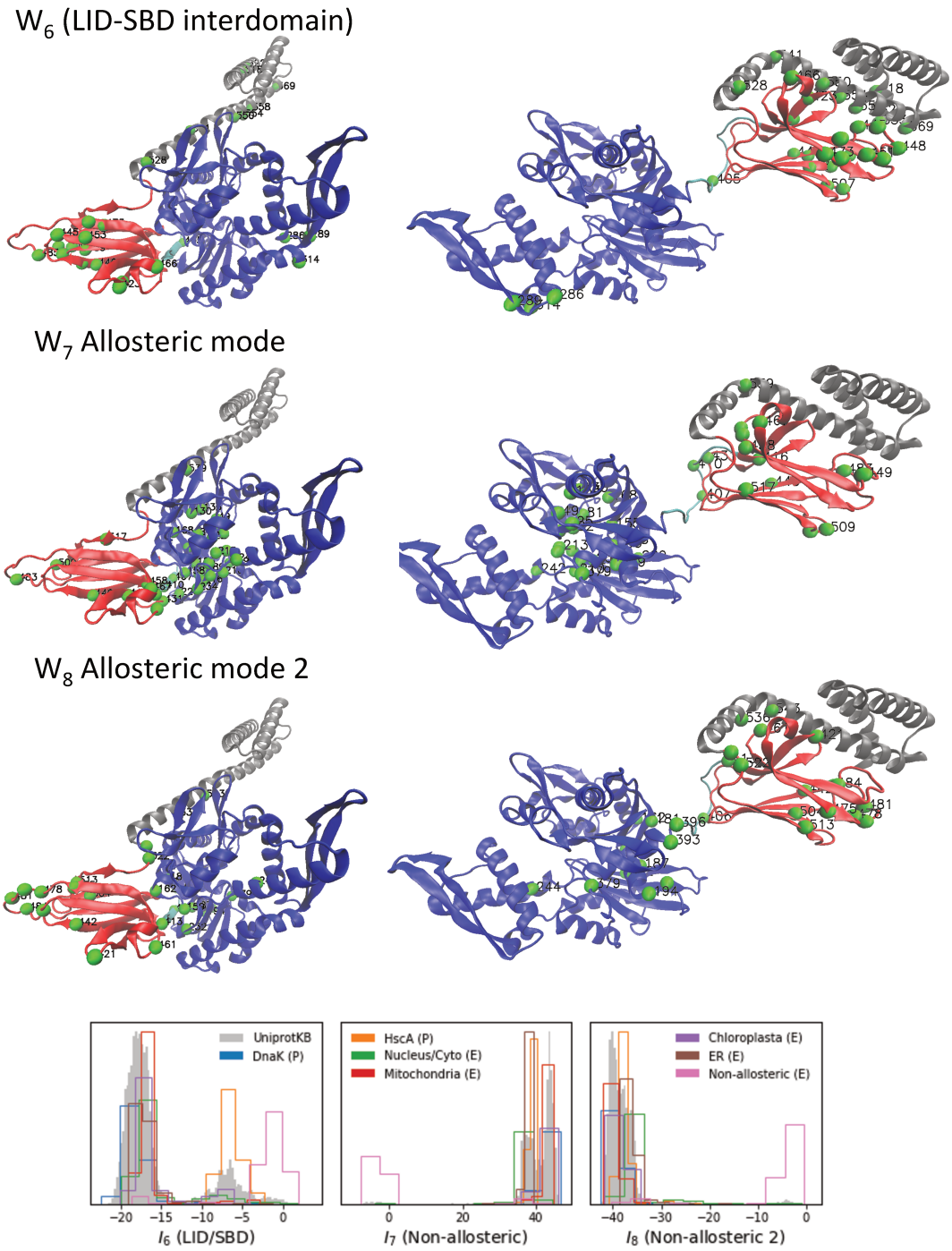

Figure 11.13: **Modeling HSP70 with RBM: Inter-domain collective modes and allostery** Structures and input distribution of weights 6,7,8 (weight logo not shown)

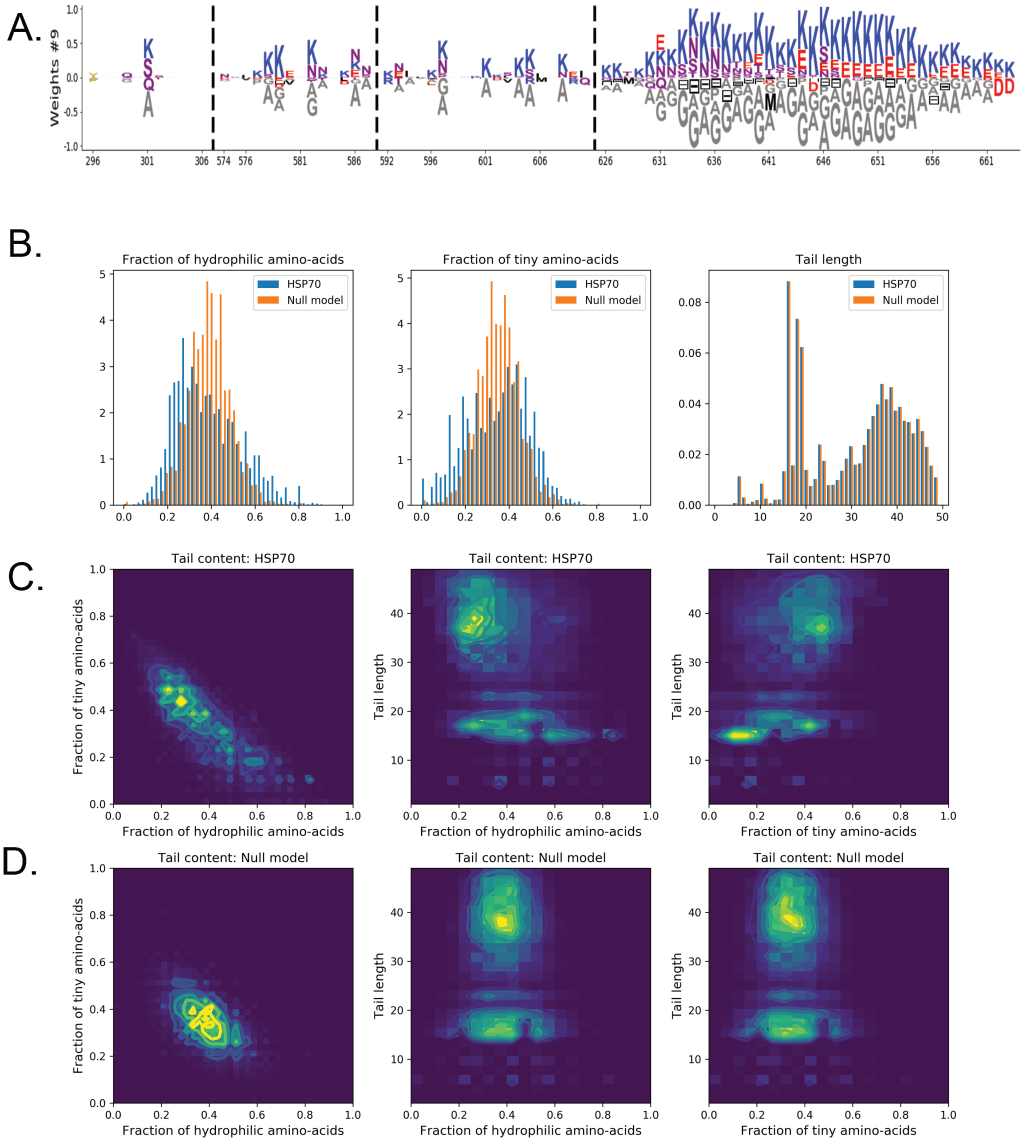

Figure 11.14: **Modeling HSP70 with RBM: Unstructured tail** A. Weight logo of weight 9. **B-D Statistics of length and amino-acid content of the tail, as compared to a null model** The distribution of the number of hydrophilic (E,D,K,R,T,S,N,Q)/tiny (A,G) amino-acid on the tail is wide and non-gaussian (B, left and middle), but this could be due to the variability in tail length (B, right). To assess the enrichment, we build a null model where the tail size is random (same statistics as Hsp70), and each amino-acid is drawn randomly, independent from the others, using the same amino-acid frequency as in the tail of Hsp70. The null model statistics (orange histograms and lower contour plots) are clearly different, validating the significance of weight 9.

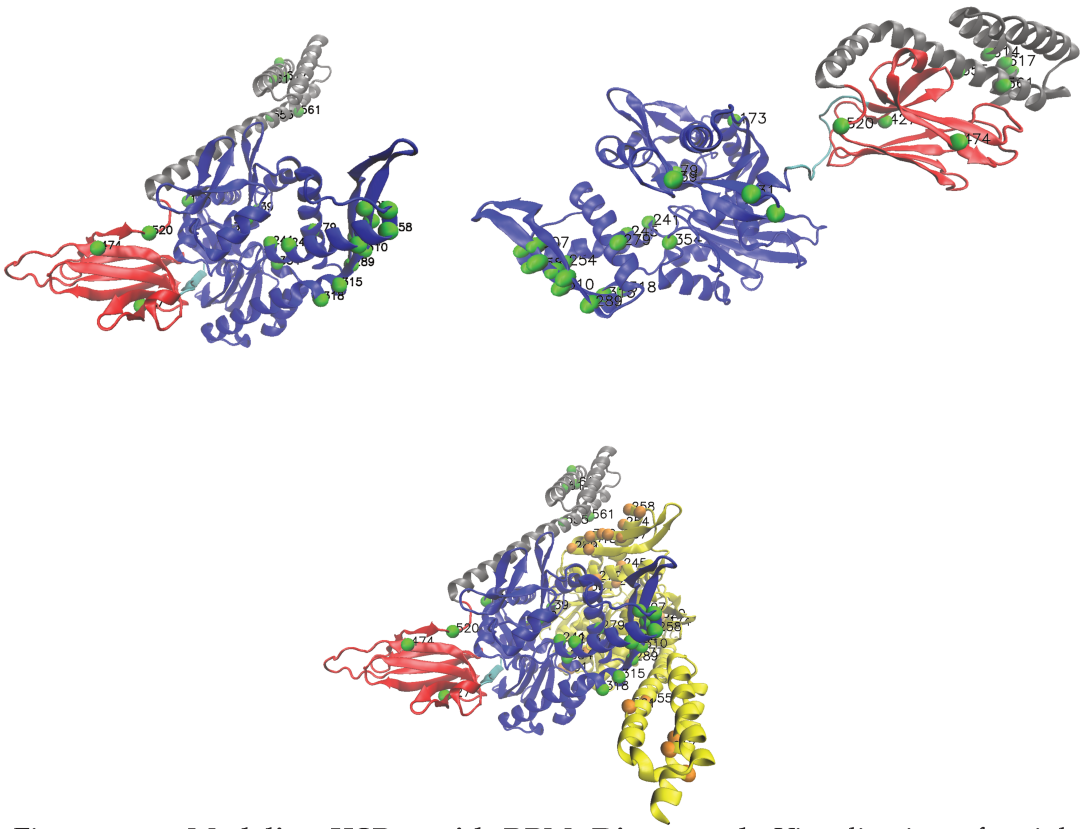

Figure 11.15: **Modeling HSP70 with RBM: Dimer mode** Visualization of weight 10 on the ATP-bound, ADP-bound structure and ATP-bound/ATP-bound dimer. In the later, the second Hsp70 is in yellow, and orange spheres denote the corresponding positions of weight 10

showed that Direct Coupling Analysis could detect conformation-specific contacts; this is similar to hidden units, respectively, 3 and 4 presented here, located on contiguous sites in the, respectively, ADP-bound and ATP-bound conformations. In [201], an inter-domain sector of sites discriminating between allosteric and non-allosteric sequences was found. This sector share many sites with our weight 4, and is also localized at the SBD/NBD edge. However, only a sector could be retrieved with sector analysis, whereas many other meaningful collective modes could be extracted using RBM.

12

CONTACT PREDICTION WITH RESTRICTED BOLTZMANN MACHINES

12.1 PRINCIPLE

As illustrated in the previous chapter, co-occurrence of large weight components in highly sparse features often correspond to nearby sites on the 3D fold. To extract structural information in a systematic way, we present here a method to derive effective pairwise interactions matrix $J_{ij}^{eff}(a,b)$ from any probability distribution over the sequence space $P(\mathbf{v})$. This matrix can then be used to estimate contacts as in direct-coupling based approaches [176]. [1]. The main idea is to estimate second-order epistasis coefficients using $P(\mathbf{v})$, then rewrite these coefficients into effective couplings. We consider a sequence $\mathbf{v}^{a,b}$ with residues a and b on, respectively, sites i and j. Single mutations $a \to a'$ or $b \to b'$ on, respectively, site i or j are accompanied by changes in the log probability of the sequence indicated by the full arrows in Fig. 12.1. Comparing the change resulting from the double mutation with the sum of the changes resulting from the two single mutations provides the model-based estimate of the epistatic interaction:

$$\Delta\Delta\mathcal{L}_{ij}(\mathbf{v};a,a',b,b') \equiv \log\left[\frac{P(\mathbf{v}^{a,b})\,P(\mathbf{v}^{a',b'})}{P(\mathbf{v}^{a',b})\,P(\mathbf{v}^{a,b'})}\right], \tag{12.1}$$

1 In the case of RBM, $P(\mathbf{v})$ is the marginal probability distribution, defined in Section II

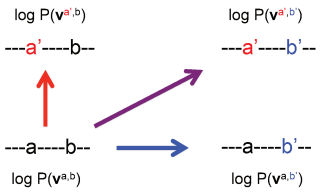

Figure 12.1: **Principle of epistatic-based contact prediction.** The change in log probability resulting from a double mutation (purple arrow) is compared to the sum of the changes accompanying the single mutations (blue and red arrows)

Applying the definition to a DCA model, with marginal distribution

$$P(\mathbf{v}) \propto \exp\left(\sum_i g_i(v_i) + \frac{1}{2}\sum_{i\neq j} J_{ij}(v_i, v_j)\right), \tag{12.2}$$

we obtain

$$\Delta\Delta\mathcal{L}_{ij}(\mathbf{v}; a, a', b, b') = J_{ij}(a, b) + J_{ij}(a', b') - J_{ij}(a, b') - J_{ij}(a', b), \tag{12.3}$$

independently of the sequence $\mathbf{v}$. With the zero-sum gauge ($\sum_b J_{ij}(a, b) = \sum_a J_{ij}(a, b) = 0$) for the couplings, equation (12.3) can be inverted, yielding:

$$J_{ij}(a, b) = \frac{1}{q^2}\sum_{a',b'} \Delta\Delta\mathcal{L}_{ij}(\mathbf{v}, a, a', b, b') \tag{12.4}$$

Thus, assuming a pairwise model implies that $\Delta\Delta\mathcal{L}ij$ is constant, i.e. independent of the background sequence, and equation (12.4) shows that the reciprocal is also true. For a general distribution $P(\mathbf{v})$, we do not expect $Delta\Delta\mathcal{L}$ to be constant as higher-order interaction terms may be present. We can nonetheless define an effective coupling matrix through:

$$J^{\text{eff}}_{ij}(a, b) = \left\langle \frac{1}{q^2}\sum_{a',b'} \Delta\Delta\mathcal{L}_{ij}(\mathbf{v}; a, a', b, b') \right\rangle_{MSA}. \tag{12.5}$$

From there, we can construct a contact map estimator based on the Frobenius norms of the effective couplings, with the Average Product Correction, see [176]. This estimator is defined for any tractable probability distribution, but it may be costly in practice, as it requires $\mathcal{O}(Bq^2N^2)$ evaluation of $P(\mathbf{v})$. In the case of RBM, each probability evaluation has complexity $\mathcal{O}(NM)$, but it is possible to reduce the complexity. Starting from the definition of $P(\mathbf{v})$ Eqn. (3.3) and writing $\tilde{I}^{ij}_\mu(\mathbf{v}) = \sum_{l\neq i,j} w_{l\mu}(v_l)$, we have:

$$\begin{aligned}
&\left\langle \Delta\Delta\mathcal{L}_{ij}(\mathbf{v}; a, a', b, b') \right\rangle_{MSA} \\
&= \sum_\mu \Big\langle \Gamma_\mu\left[\tilde{I}^{ij}_\mu(\mathbf{v}) + w_{\mu i}(a) + w_{\mu j}(b)\right] + \Gamma_\mu\left[\tilde{I}^{ij}_\mu(\mathbf{v}) + w_{\mu i}(a') + w_{\mu j}(b')\right] \\
&-\Gamma_\mu\left[\tilde{I}^{ij}_\mu(\mathbf{v}) - w_{\mu i}(a) + w_{\mu j}(b')\right] - \Gamma_\mu\left[\tilde{I}^{ij}_\mu(\mathbf{v}) - w_{\mu i}(a') + w_{\mu j}(b)\right] \Big\rangle_{MSA}
\end{aligned} \tag{12.6}$$

Since the expression is a sum over hidden units, it involves only marginal statistics of $\tilde{I}^{ij}_\mu(\mathbf{v})$. For a fixed μ, i, j, one can replace the $\langle . \rangle_{MSA}$ by an average

over the distribution of $\tilde{I}_\mu^{ij}(\mathbf{v})$, which can be approximated with an histogram of, say, $n_{bins} = 100$ bins (total cost $\mathcal{O}(MN^3B)$). Then, we scan through a, b and compute the q^2 coefficients. The overall cost is therefore $\mathcal{O}(N^3MB + n_{bins}N^2Mq^2)$, instead of $\mathcal{O}(N^3MBq^2)$.

A fast approximation can also be derived by writing a second-order Taylor expansion of Γ_μ in Eqn. (12.6). After rearrangement, we obtain:

$$J_{ij}^{eff}(a,b) = \frac{1}{2}\sum_\mu w_{i\mu}w_{j\mu}\left\langle \Gamma_\mu''(\mathbf{v})\right\rangle_{MSA} \tag{12.7}$$

The Taylor expansion is exact when Γ_μ'' (The conditional variance) is constant, *i.e.* for quadratic potential, and we recover exactly the original expression of the couplings of Eqn. (3.6). For a non-quadratic potential, this equation illustrate the dependency of the coupling with the sequence. In particular, an 'inactive' hidden unit, i.e. such that I_μ lies in a saturation of the average activity Γ_μ' does not produce any epistatic effect around the sequence.

Overall, this estimator of contacts is fairly natural and its definition coincides with the ones of pairwise models. For non-pairwise models, we note that other averaging schemes could be investigated, such as computing quantiles rather than average.

12.2 RESULTS

12.2.1 *Contact prediction*

We use the above method to derive effective couplings and predict contact maps from the RBM trained on the Kunitz Domain, WW domain and lattice protein families shown in Section III. The main results are summarized in Figure 12.2. Panels A-E focus on the Kunitz domain; panel A,B illustrate how the true contact map is faithfully reproduced by the estimator based on RBM. From a quantitative point of view, the Positive Predicted Value curves of predicted contacts (Panel C) and distant contacts (panel D) show comparable performance as contact map prediced using DCA, trained either via pseudo-likelihood maximization (PLM, [115]) or Monte Carlo learning (BM). Moreover, the effective couplings of Eqn. (12.5) correlate well with the ones inferred by DCA, see panel E. Similar performance and behavior are found for WW and LP.

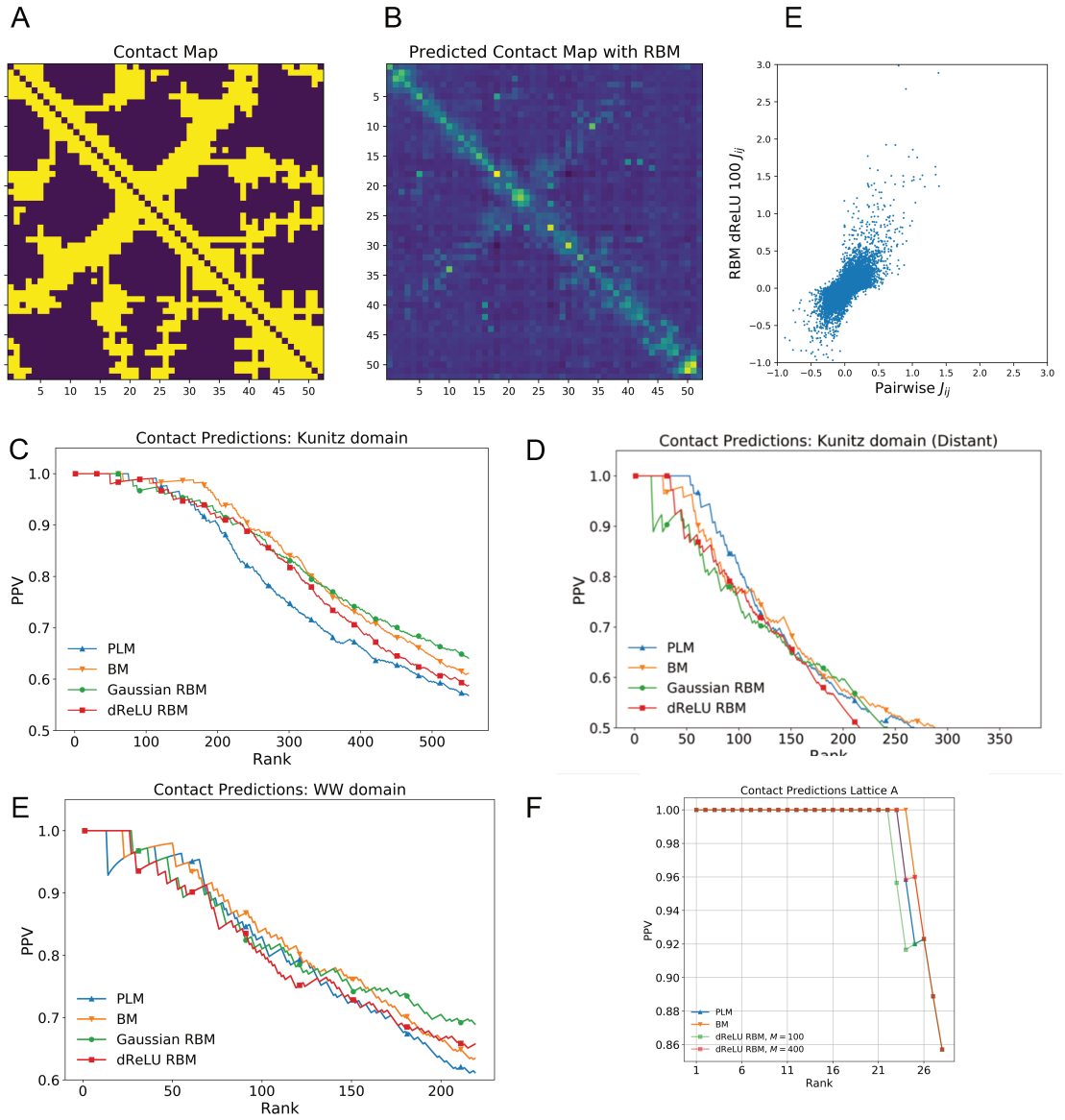

Figure 12.2: **Contact predictions using RBM, applied to Kunitz domain (A-E), WW domain (F) and Lattice Proteins (G)** . **A** Binary contact map of the Kunitz domain, based on PDB structure 2knt [229]. It equals 1 if two residues i, j have a distance between their C_α atoms shorter than 8 Å. **B** Predicted contact map using RBM. **C,D** Fraction of correctly predicted contacts against the number of predicted contacts, for all contacts (**C**) or distant contact only ($|i - j| > 4$, **D**). **E** Scatter plot of pairwise couplings inferred by DCA against effective couplings inferred by RBM. **F,G** Same as panel **C** for the WW domain and Lattice Proteins.

12.2.2 *Dependence on the parameters of the RBM*

We now assess how the quality of contact predictions depends with the parameters of the RBM: hidden-unit potential, number of hidden units and regularization choice. We repeat the contact predictions process on the three protein families with various parameters and show main results in Fig. 12.3. We find that the quality of predictions:

- strongly increases with the number of hidden units. This dependence is not surprising, as the number M of hidden units acts in practice as a regularizor over the effective coupling matrix between residues. In the case of Gaussian RBM, the value of M fixes the maximal rank of the matrix $J_{ij}(v_i, v_j)$. The value $M = 100$ of the number of hidden units is small compared to the maximal ranks $R = 20 \times N$ of the couplings matrices of the Kunitz ($R = 1060$) and WW ($R = 620$) domains, and explains why Direct-Coupling Analysis gives slightly better performance than RBM in the contact predictions of Fig. 12.2.

- (i) is slightly better for quadratic and dReLU potentials than Bernoulli potentials, and (ii) there is little to no difference between quadratic and dReLU potentials. It is somewhat expected that Bernoulli potentials perform less at fixed $M << R$, as they are less expressive than Gaussian and dReLU potentials. On the other hand, the fact that Gaussian and dReLU potentials perform almost the same, despite strongly non-linear activation functions and different generative performance (see Section V) is more puzzling, and requires further investigation.

- tends to improve with the weight sparsity, see panel E and F. We indeed expect small regularization to improve contact predictions as it prevents overfitting; it is the case in pseudo-likelihood maximization for instance. We note that stronger regularization seem to slighly improve performance as well (upper left corner of panels E,F), and it would be interesting to investigate why.

12.2.3 *Conclusion*

Overall, it is possible to exploit RBM for contact prediction purposes, and we can reach performance equivalent to pairwise couplings methods for small protein families. We note however that the larger the protein, the larger the number of hidden units required to reach the performance of pairwise model.

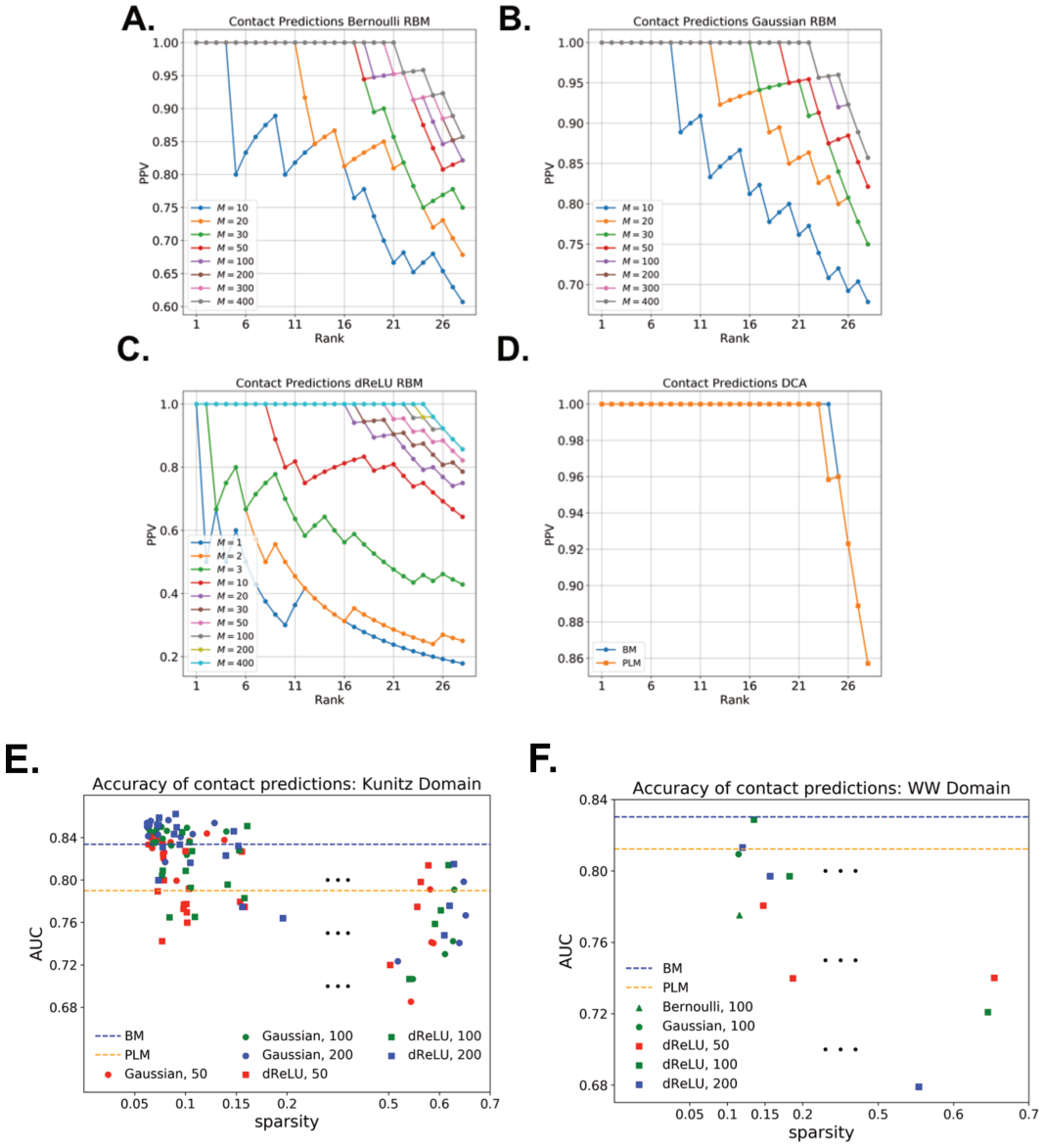

Figure 12.3: **Dependence of contact predictions on the RBM parameters (A-D)** Positive Predicted Values plots for Lattice Proteins, with Bernoulli **(A)**, Quadratic **(B)** and dReLU **(C)** potentials and pairwise models **(D)**. **(E) Kunitz, (F) WW** Area under curve metric (defined as the integral of the PPV - Rank curve, integrated up to the true number of contacts) for various models with different training parameters, regularization choice and hidden units number/potentials, against the weight sparsity.

Therefore, using RBM for contact prediction shows little speed gain compared to Boltzmann Machine Learning in practice. In the absence of an efficient approximate inference algorithm, it is currently preferable to use other standard algorithm such as plmDCA. Future investigation for the application of RBM for contact prediction include the design of more efficient learning algorithm and different effective coupling computation (e.g. different averaging schemes, selecting only subset of hidden units,...).

13

PROTEIN DESIGN WITH RESTRICTED BOLTZMANN MACHINES

As discussed in Section 9.2.2, generative models like BM and RBM can be used to score sequences and generate artificial sequences with putative natural-like structure and function. However, as illustrated in Section 11, several protein families feature a diversity of functional specializations: substrate specificity, protein partners, biological expression,... Could we specify in advance what is the functionality of these sequences ? This is particularly important for achieving controlled protein design. Similarly, an ideal theoretical fitness function should take into account the specific details of an experiment: nature of the substrate, experimental pH,... How can we modify the statistical energy function in order to take into account these information ? Here, we show how the biologically interpretable representation learnt by RBM can guide quantitatively protein design and scoring. Beyond designing natural-like sequences, we illustrate how RBM can generate sequences in regions of the sequence space not seen in the alignment. Such approaches could provide rationales for better understanding the necessary conditions to functionality, and designing proteins with non-natural properties.

13.1 METHODS OF BIASED SAMPLING

13.1.1 *Conditional sampling*

We have shown in Section 11 that several hidden units reliably identified functional subgroups within a protein family. In the context of design, a natural way to leverage this property is to sample while fixing the value of these hidden units. Numerical implementation of conditional sampling is straightforward in

RBMs. For instance, the distribution of sequences $\mathbf{v}$ conditioned hidden unit μ having activity equal to h_μ^c gives

$$\begin{aligned} & P(\mathbf{v}|h_\mu = h_\mu^c) \\ & = \frac{1}{P(h_\mu = h_\mu^c)} \int \prod_{\nu(\neq\mu)} dh_\nu \, P(\mathbf{v}, \mathbf{h}) \\ & \propto \exp\left[\sum_i g_i(v_i) - \sum_{\nu(\neq\mu)} \mathcal{U}_\nu(h_\nu) + \sum_{\nu(\neq\mu),i} w_{i\nu}(v_i) h_\nu + \sum_i w_{i\mu}(v_i) h_\mu^c \right] \end{aligned} \tag{13.1}$$

which is formally the probability distribution of another RBM with N visible units, $M-1$ hidden units, visible layer fields $\tilde{g}_i(v) = g_i(v) + w_{i\mu}(v) h_\mu^c$ and identical weights and potentials $\tilde{w}_{i\nu}(a) = w_{i\nu}(a)$, $\tilde{\mathcal{U}}_\nu = \mathcal{U}_\nu$, for all $\nu \neq \mu$. Conditioning is therefore equivalent to removing the hidden unit and multiplying the distribution by a factor exponential in the input $I_\mu(\mathbf{v})$. More generally, simultaneous conditioning over K hidden units yields an RBM with $M-K$ hidden units. Such conditioned RBM model can be used either for sampling or scoring.

13.1.2 *Low temperature sampling*

Traditional sampling of $P(\mathbf{v})$ produces artificial sequences with average statistical energy. To increase the chance of finding sequences with high statistical fitness, one important trick is to bias sampling toward sequences having low statistical energy. For a traditional exponential model such as Boltzmann Machines, this is achieved by low temperature sampling. We define the modified distribution $P_\beta(\mathbf{v}) = \frac{e^{-\beta E(\mathbf{v})}}{Z_\beta}$, where β is the inverse temperature. For instance, $P_2(\mathbf{v}) \propto P_1^2(\mathbf{v})$, thus reducing the importance of low probability sequences. In the case of RBM, biased sampling is not straightforward, as the low temperature of the marginal distribution $P_\beta(\mathbf{v})$ is not in general the marginal of the low temperature of the joint probability $P_\beta(\mathbf{v}, \mathbf{h})$. The trick is to duplicate the hidden units, the weights, and the local potentials acting on the visible units, as shown in Fig. 13.1. By doing so, the sequences $\mathbf{v}$ are distributed according to:

$$P_2(\mathbf{v}) \propto \int \prod_\mu dh_{\mu 1} \, dh_{\mu 2} \; P(\mathbf{v}|\mathbf{h}_1) \, P(\mathbf{v}|\mathbf{h}_2) = P(\mathbf{v})^2 \, . \tag{13.2}$$

In other words, sampling from $P_\beta(\mathbf{v})$ for integer $\beta > 1$ is possible, and done by duplicating β times the hidden layer and visible layer fields.

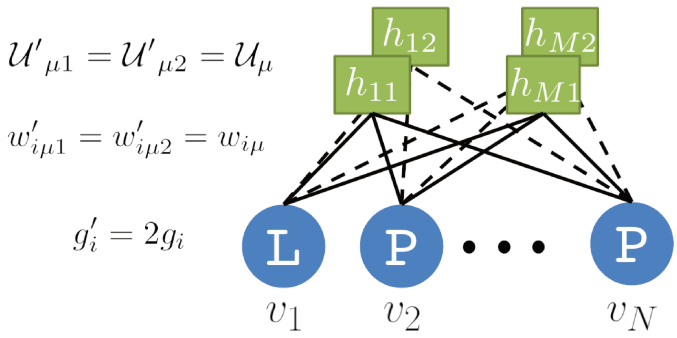

Figure 13.1: Duplicate RBM for biasing sampling toward high-probability sequences. Visible-unit configurations $\mathbf{v}$ are sampled from $P_2(\mathbf{v}) \propto P(\mathbf{v})^2$.

13.1.3 *Focused sampling*

One last useful tool is to bias sampling around a known experimental sequence $\mathbf{v}^{WT}$. This is done by adding a field $\delta g_i(v) = \lambda \delta_{v,v_i^{WT}}$, where $\lambda \geq 0$ is a Lagrange multiplier. Larger λ allows to sample closer to $\mathbf{v}^{WT}$ in terms of Hamming distance.

13.2 RESULTS

13.2.1 *Conditional Sampling and feature recombination*

The biological interpretation of the features inferred by the RBM guides us to sample new sequences $\mathbf{v}$ with putative functionalities. For WW domains, we condition on the activities of hidden units 4 and 5, related to binding specificity, see Fig. 11.3. Fixing h_4 and h_5 to levels corresponding to the peaks in the histograms of inputs in Fig. 11.3C allows us to generate sequences belonging specifically to each one of the three ligand-specificity clusters, see Fig. 13.2A. In addition, sequences with combinations of activities that are not encountered in the natural MSA can be engineered. As an illustration, we generate by conditional sampling hybrid WW-domain sequences with strongly negative values of h_3 and h_4, corresponding to a Type I-like β_2-β_3 binding pocket and a long, Type IV-like β_1-β_2 loop (red dots in panel A and corresponding sequence logo D). For Kunitz domains, the property 'no 11-35 disulfide bond' holds only for some sequences of nematode organisms, whereas the Bikunin-AMBP gene

is present only in vertebrates; they are thus never observed simultaneously in natural sequences. Sampling our RBM conditioned to appropriate levels of h_2 and h_5 allows us to generate sequences with both features activated (red dots in panel B and corresponding sequence logo E). In Lattice Proteins, the sampling is ergodic but the MSA is of finite size. Hidden units 3 and 4 of RBM shown in Fig. 11.2 are independent, but both have very sparse activity, such that we never observe a sequence with both strong activations. RBM can generate sequences having both activities (blue and cyan dots, panel C).

13.2.2 *The fitness - diversity trade-off*

A good generative model must be able to generate sequences that have both high fitness and high diversity, *i.e.* sequences that are far away from one another and from the training set sequences. Indeed, since RBM are universal approximators (see Section 3.2), they could very well overfit the training set, such that samples are copies of the original sequences (up to a few quasi-neutral mutations). As shown in panels A and B of Fig. 13.3, the sequences designed by RBM are far away from all natural sequences in the MSA, but have comparable probabilities. The sequences are also compatible with a DCA model trained on the same data (panels C and D). The general trend is that the farther away from natural sequences, the lower the likelihood - this is expected. However, we can also find high likelihood sequences that are significantly different from natural sequences using low temperature sampling. Interestingly, sequences generated by conditional sampling with unseen combination also have high likelihood, despite never being seen in the data. This extrapolation is directly related to the compositional phase: recombining compatible, non-overlapping features yields sequences with similar likelihood as regular ones.

The capability of RBM to design new sequences with desired features and high values of fitness can be validated on Lattice Proteins, as the fitness function is well-defined in this case. This was previously done for BM in [205]. Figure 13.3E shows that the log-likelihood correlates well with the fitness P_{nat}, both for training and test set sequences. Sequences designed by RBM are diverse and have high P_{nat} (panel F); in particular those designed by combining h_3 and h_4. Remarkably, low temperature sampling allowed to find several sequences with higher P_{nat} than the highest value in the training MSA.

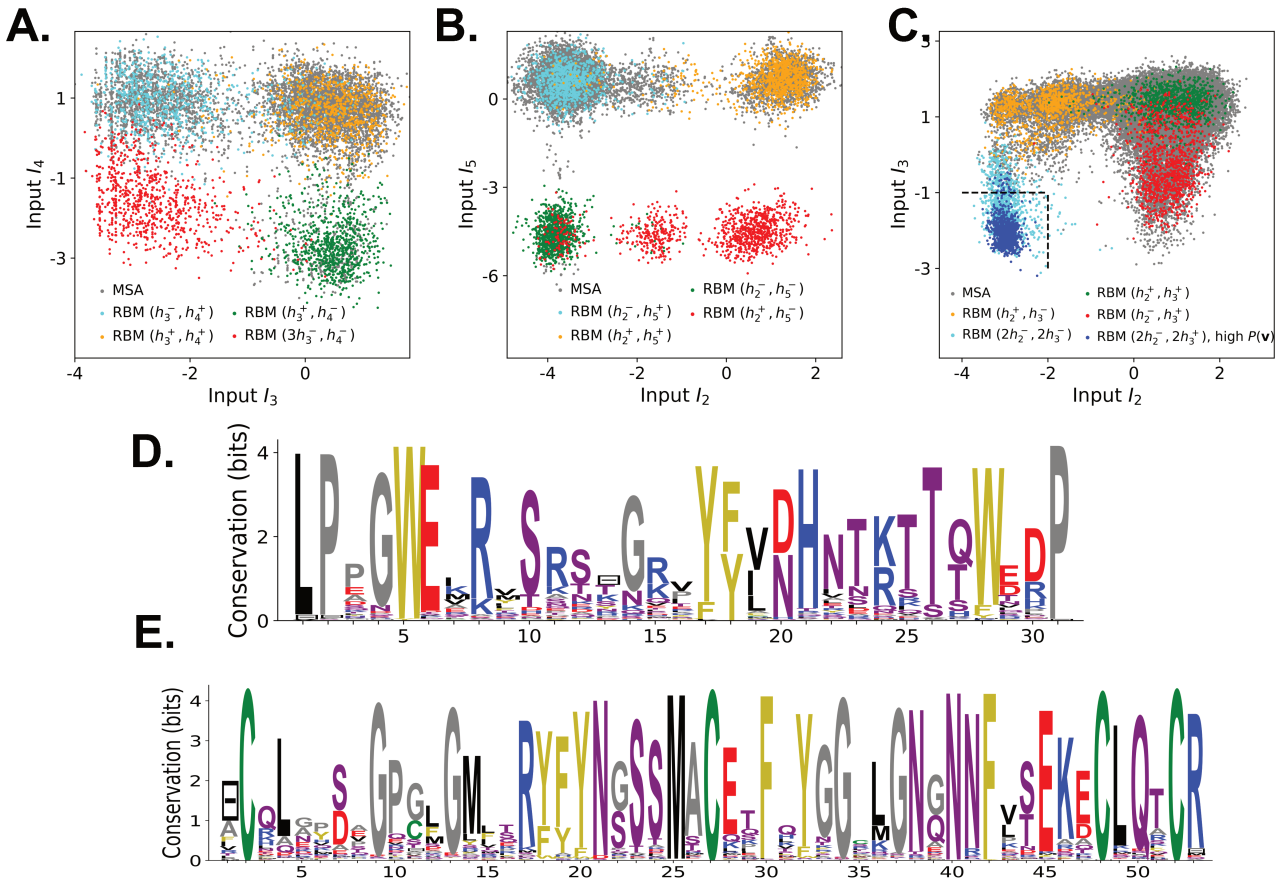

Figure 13.2: **Sequence design with RBM. A.** Conditional sampling of WW domain-modeling RBM. Sequences are with activities (h_4, h_5) fixed to (h_4^-, h_4^+), (h_4^+, h_5^-), (h_4^+, h_5^+) and $(3h_4^-, h_5^-)$ see red points indicating $h_4^\pm, h_5^\pm$ in Fig. 11.3 C. Natural sequences in the MSA are shown with gray dots, and generated sequences with colored dots. **B.** Conditional sampling of Kunitz domain-modeling RBM, with activities (h_2, h_6) fixed to $(h_2^\pm, h_6^\pm)$, see red dots indicating $h_2^\pm, h_6^\pm$ in Fig. 11.5 **C.** Similar conditional sampling of Lattice Proteins RBM. **D** Sequence logo of the red sequences in panel A, with 'long β_1-β_2 loop' and 'type I' features. **E.** Sequence logo of the red sequences in panel B, with 'no disulfide bridge' and 'bikunin' features.

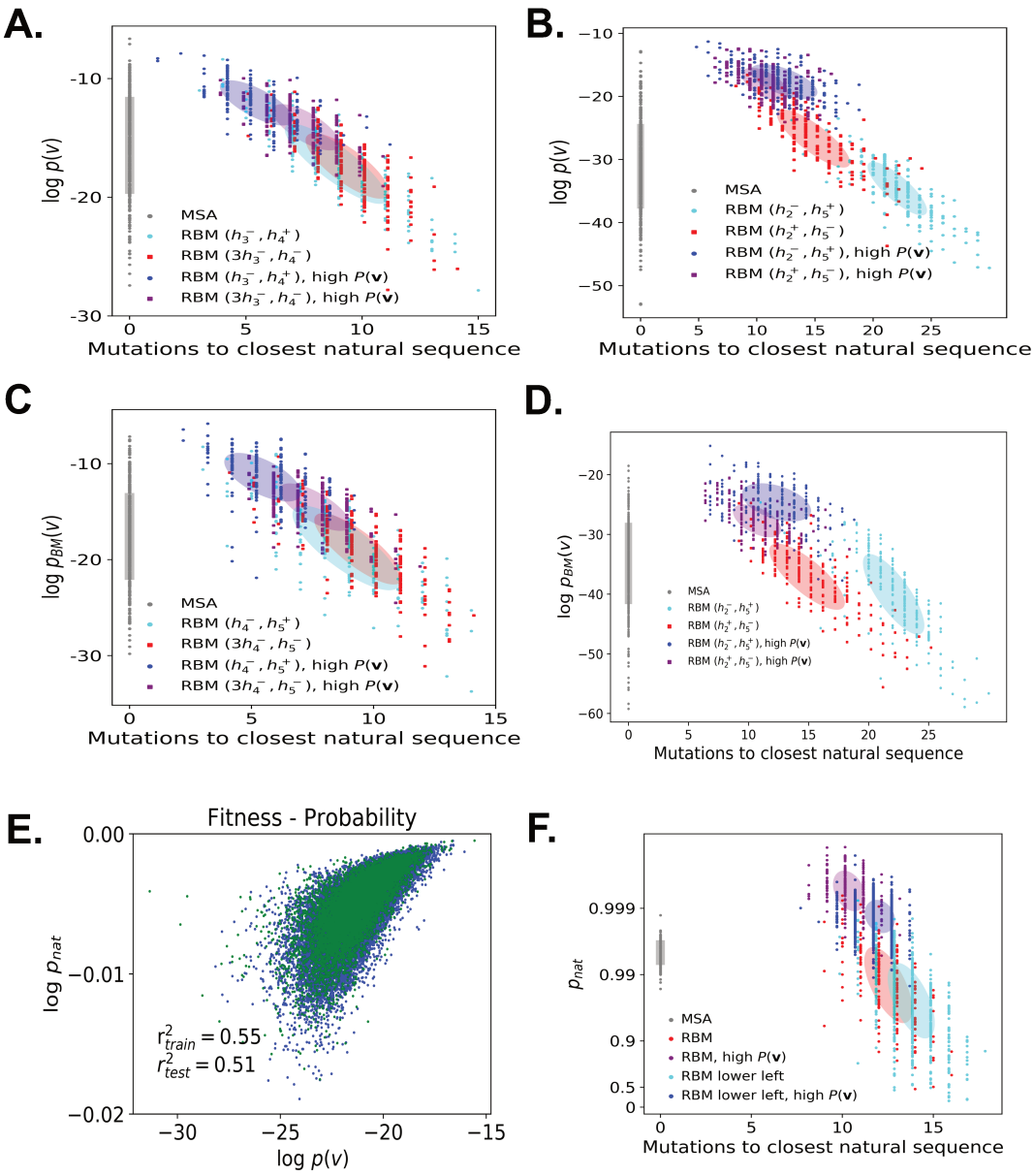

Figure 13.3: **Sequence design with RBM. A.** Scatter plot of the number of mutations to the closest natural sequence vs log-probability, for natural (gray) and artificial (colored) WW domain sequences. Same color code as panel 13.2A; dark dots were generated with the low temperature sampling.**B** Same as panel A, but with the log-probability of a pairwise model trained on the same data. **C,D** Same for Kunitz domain. **E** Log-probability of Lattice Protein RBM vs true fitness evaluated on sequences from the MSA used (train) or not (test) for training. **F** Scatter plot of the number of mutations to the closest natural sequence vs the Lattice Protein fitness P_{nat} for RBM-generated sequences.

13.2.3 *Converting protein specificities*

Beyond the determination of specificity-determining positions within a protein family, one important question is to find plausible evolutionary paths between subfamilies. Consider the case of two protein subfamilies, such as the serine protease family, and trypsin/chymotrypsin-specificity, with a set of associated specificity-determining positions (SDP) $S = \{s_l\}$, with amino-acid a_l (for the first family), a'_l (for the second family). Sequences having $v_{s_l} = a_l$ likely functionally belong to the first family, and conversely. Given a sequence $\mathbf{v_1}$ belonging to the first family, we ask what sequence of mutations should we perform in order to bring it to the second subfamily. We should at least mutate the SDPs, but in which order, so as to maintain some functional fitness ? Moreover, is mutating only the SDPs enough, or additional mutations are required to restore structural stability ? We show in a simple scenario that RBM may provide an answer. In the case of trypsin, we found that a single hidden unit, h_1 differentiated trypsin-type specificity ($I_1 \ll 0$) from chymotrypsin-type specificity ($I_1 > 0$), see Fig. 11.8 and Fig. 11.9). As seen from the weight logo, this amounts to about 18 SDPs. Given the wild type (say, rat chymotrypsin), we define a conditional focused RBM, by conditioning the RBM on h_1 (value h_1^c) and focusing it around the wild type (with lagrange parameter λ). In the limit where $\lambda \to \infty$, all samples collapse on the wild type. Conversely, when $h_1 \to +\infty$ and $\lambda = 0$, we obtain traditional samples with $I_1 > 0$, such as the rat trypsin. For intermediate value of h_1 and λ, low temperature samples of the conditional focused RBM are gradually farther away from the wild type, while going from $I_1 \ll 0$ to $I_1 > 0$. Scanning through values of h_1, λ allows to find low energy transition paths connecting the wild type and the $I_1 \gg 0$ subspace, putatively corresponding to chymotrypsin specificity, see Fig. 13.4. Compared to simply switching the SDPs amino-acids, the path found by RBM have a relatively low (statistical) energetic cost. We are currently collaborating with the Statistical Biology group of Rivoire and Nizak at College de France, and look forward for experimental tests of these transition paths. More generally, finding a path between two sequences or subspaces involves more than a single hidden unit switch, and more general transition path sampling techniques should be developed in the future. It would be very interesting to estimate the number of possible transition path between two proteins, as was partially done experimentally by Poelwijk et al. between two fluorescent proteins [197].

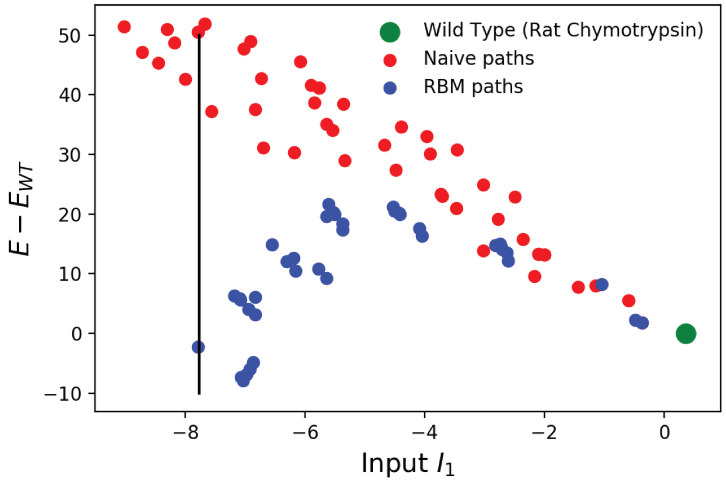

Figure 13.4: **Protein specificity conversion with RBM.** Scatter plot of the hidden input I_1 to the RBM free energy, for sequences gradually mutated from the rat chymotrypsin. Black line marks the hidden input value of rat trypsin. Green dot denotes the wild type, Red dots denote naive specificity conversion (Mutating in random order the SDPs), while blue dots are proposed RBM conversion path

14

MODEL SELECTION

We presented in the previous sections various results from trained RBM, without justification for the model parameters chosen (strength of regularization, number of hidden units, shape of hidden-unit potentials, ...). We motivate these choices a posteriori in this chapter, based on model performance and interpretability.

Here, performance is measured by the accuracy of the fit of the model distribution to the empirical data distribution. It is evaluated by the average log-likelihood, divided by the number of visible units $\frac{1}{N}\langle \log P(\mathbf{v}) \rangle_{MSA}$ on a train set - to assess the capacity of the model and on a held-out test set, not used for training, to assess the ability to generalize. For visible-unit variables with $q = 21$ possible values (20 amino acids + gap symbol), this number typically ranges from $-\log 21 \simeq -3.04$ (uniform distribution) to 0. Evaluating the likelihood requires knowledge of the partition function, see Part I Section 3.5. We acknowledge that log-likelihood is not the ultimate metric of model performance: in the context of sequence design and scoring, that would be the quality/diversity of generated sequences or the correlation with the true fitness landscape. However, we found a good correlation between log-likelihood and sequence quality in the case of Lattice Proteins, see Fig. 14.1, which justifies model selection based on this criteria.

We say that a model is interpretable when (i) there is a simple link between weight matrices and typical configurations from the data or model distribution, and (ii) weights can be easily related to the biological constraints underlying protein structure and function. A simple weights - configuration relationship is achieved in the Random-RBM model with sparse weights, under the compositional phase introduced in Part iii. A typical hidden layer configuration consists in L strongly activated hidden units, and the rest are silent, and the corresponding visible layer configuration have high overlaps with the L selected weights. Since weights do not overlap, all combinations are possible, and each sequence can be mapped into one such combination, as in Fig. 2.8. Of course, the Random-RBM model is a poor depiction of real protein fitness landscape: N is small and the effective temperature can be fairly high, such that there is not always a nice scale separation between active and inactive hidden units, and

importantly, features are overlapping. We will assess under which conditions the inferred model tends to the Random-RBM case.

14.1 GENERATIVE PERFORMANCE

14.1.1 *Number of hidden units*

The number of hidden units is critical for the generative performance. We trained RBMs on the Lattice Protein and WW data set for various potentials (Bernoulli, quadratic and dReLU), number of hidden units (1-400) and regularizations ($\lambda_1^2 = 0$, $\lambda_1^2 = 0.025$,$\lambda_1^2 = 0.25$). The likelihood estimation shows that, as expected, the larger M, the better the ability to fit the training data, see Fig. 14.1. Overfitting *i.e.* a decrease in test set performance may occur for large M, low regularization and/or for low B.

14.1.2 *Hidden-unit potentials*

A priori, the major difference between Bernoulli, quadratic and dReLU potentials are that (i) Bernoulli hidden unit take discrete values whereas quadratic and dReLU take continuous ones and (ii) After marginalization, quadratic potentials create pairwise effective interactions whereas Bernoulli and dReLU create non-pairwise ones. It was shown in the context of image processing and text mining that non-pairwise models are more efficient in practice, and theoretical arguments also highlight the importance of high-order interactions, see Partiii. In terms of generative performance, the above numerical experiments on Lattice Proteins and WW domain MSAs show that at equal number of parameters, dReLU RBM perform better than Gaussian and Bernoulli RBM. Similar results, not shown, were obtained for the Kunitz domain MSA. Although RBM with Bernoulli hidden units are known to be universal approximators as $M \to \infty$ [95], they require more hidden units than the other types; hence more data. This can be intuitively explained by the fact that Bernoulli units cannot naturally express modulation in the degree of presence of a feature. To overcome this issue, one needs more than one hidden unit to encode each feature, as in [93]. This is consistent with the heavier distribution of hidden units correlations observed in all data sets, see Fig. 14.2. For example, for RBM for Bernoulli potentials, 51 out of 100 hidden units encode gap stretches, as opposed to 23 for quadratic and 15 for dReLU potentials; on WW, the numbers are respectively 18, 15 and 9. For both data sets, dReLU encode more efficiently the gap modes.

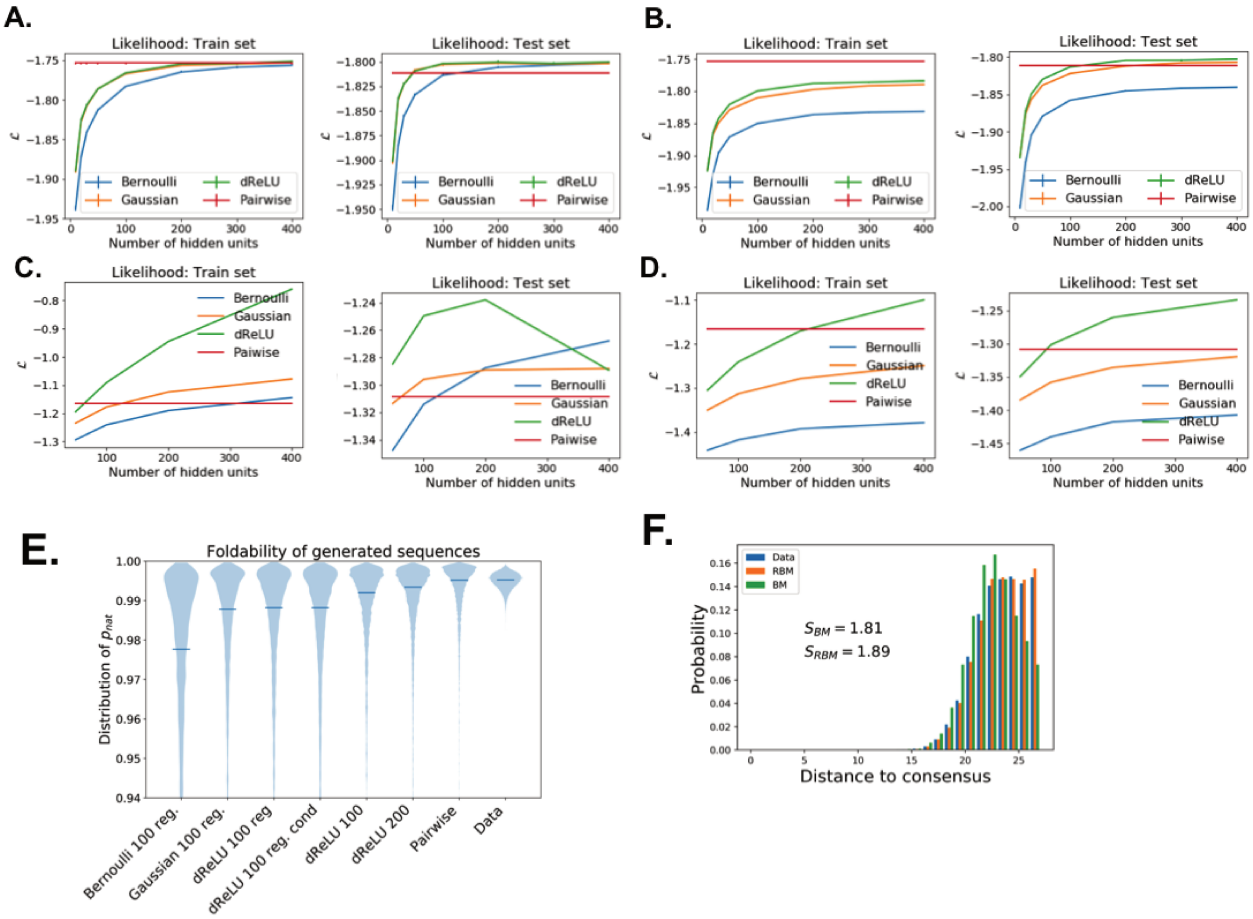

Figure 14.1: Model selection for RBM trained on LP and WW. Top, Middle: Likelihood estimates for various potentials and number of hidden units, evaluated on train and held out test set. Top: LP. Middle: WW. Left: without regularization ($\lambda_1^2 = 0$). Right: With regularization ($\lambda_1^2 = 0.025/0.25$). Bottom Left: Distributions of fitness p_{nat}, for LP sequences generated by various models. Bottom Right: Distribution of distances from a randomly selected wild type, and entropy. The quality and diversity of sequences generated correlates well with the model log-likelihood

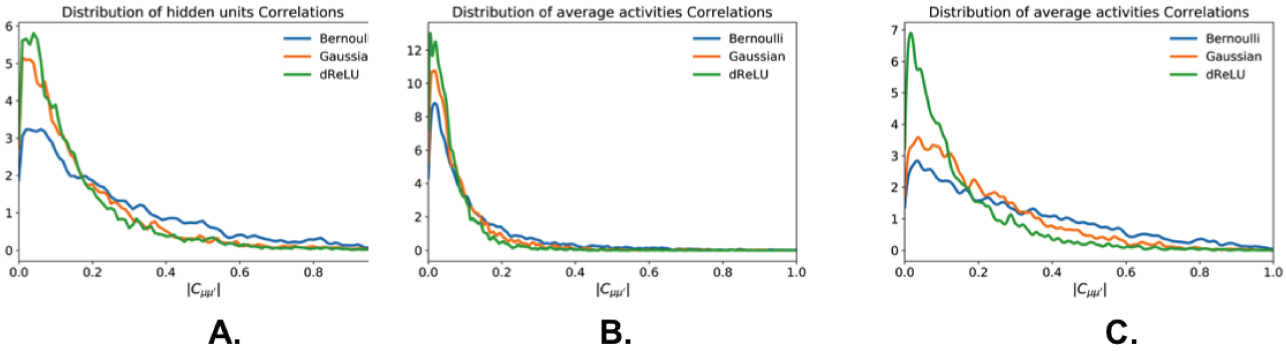

Figure 14.2: Hidden layer representation redundancy as function of the hidden-unit potentials. Distribution of Pearson correlations coeffcients between hidden-unit average activities, for RBM trained with $M = 100$, on (a) Lattice Proteins MSA, (b) Kunitz domain MSA, (c) WW domain MSA. Bernoulli RBM feature higher correlations

One of the key aspect that explains the difference of performance between dReLU and Gaussian RBM is the ability of the former to better model 'outlier' sequences, with rare extended features such as Bikunin-AMBP in Kunitz domain (Weight 6 in Fig. 11.5) or non-aromatic W28-substitution feature (Weight 3 in Fig. 11.4). Indeed, thanks to the thresholding effect of the average activity, dReLU can account for outliers, without altering the distribution for the bulk of other sequences - unlike quadratic potentials. To illustrate this idea, we compare in Fig. 14.3 the likelihoods for all sequences of two RBMs trained with quadratic (resp. dReLU) potentials, $M = 100$, $\lambda_1^2 = 0.25$ on the Kunitz domain MSA. The color code measures the degree of anomaly of the sequence, which is obtained as follows:

(a) Compute average activity h_μ^l of dReLU RBM for all data sequences $\mathbf{v}^l$,

(b) Normalize (z-score) each dimension: $\hat{h}_\mu = \frac{h_\mu - \langle h_\mu \rangle_{MSA}}{\sqrt{\text{Var}[h_\mu]_{MSA}}}$,

(c) Define:

$$c^l = \arg\max_\mu |\hat{h}_\mu^l| \tag{14.1}$$

For instance, a sequence $\mathbf{v}^l$ with $c^l = 10$ has at least one hidden-unit average activity that is 10 standard deviations away from the mean. Clearly, most sequences have very similar likelihood but the outlier sequences are better modeled by dReLU potentials.

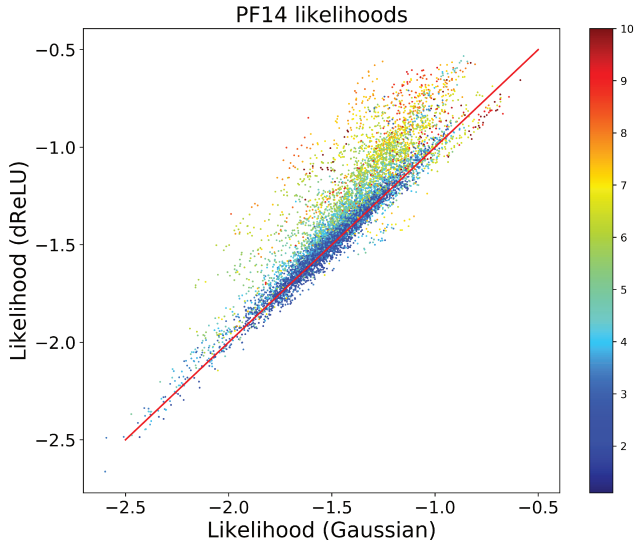

Figure 14.3: Comparison of Gaussian and dReLU RBM with $M = 100$ trained on the Kunitz domain MSA. Scatter plot of likelihoods for each model, where each point represents a sequence of the MSA. The color code is defined in Eqn. 14.1; hot colors indicate 'outlier' sequences

14.1.3 *Sparse regularization*

Weight Sparsity, which is crucial for both interpretability and compositionality, does not arise naturally from training RBM on protein sequences; a sparse penalty term is required. We have introduced the L_1^2 penalty $\propto \frac{1}{2Nq}\sum_\mu \left(\sum_{i,v}|w_{i\mu}(v)|\right)^2$, and we will compare it now to the L_1 penalty $\propto \sum_{\mu,i,v}|w_{i\mu}(v)|$. As seen from Figs. 14.4 (Lattice Proteins) and (WW), sparsity is a standard regularizer: high penalties produce a sharp decrease of the training set likelihood, but only a mild decrease, if not an increase of the test set likelihood. For both L_1 and L_1^2 regularizations, sparsifying largely reduces the effective number of parameters, which is crucial for modeling small data sets. We note however that for L_1 regularization, several hidden units become disconnected (*i.e.* $w_{i\mu}(v) = 0$ for all i, v) as we increase the penalty strength (Fig. 14.4 E). Moreover, the likelihood drops more abruptly for the L_1 penalty than the L_1^2, which makes model selection harder. Overall, the L_1^2 penalty achieves sparse weights without disconnecting hidden units when the penalty is too large, hence it is more robust and requires less fine tuning.

In practice, the values of λ_1^2 maximizing the test set log-likelihood produce weights that are not really sparse, see Fig. 14.5. To pick a regularization value in

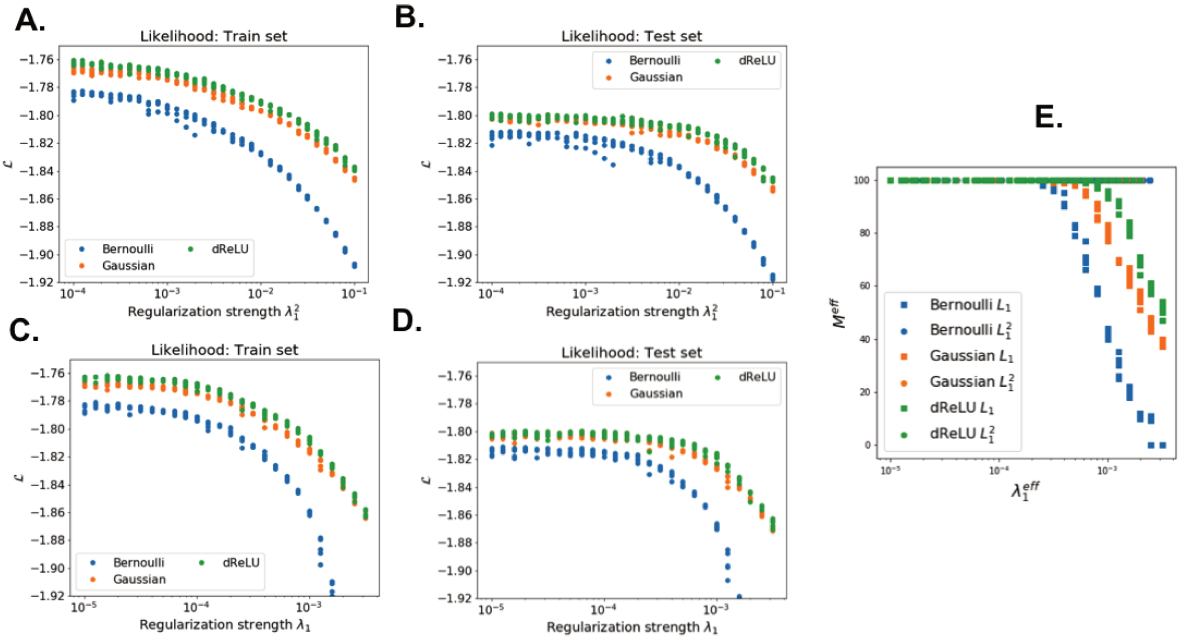

Figure 14.4: Role of regularization for RBM trained on the MSA of the Lattice Protein S_A. Panels A-D: Likelihood as function of regularization strength, for L_1^2 (top) and L_1 (bottom) sparse penalties, on train (left) and test (middle) sets. E: Number M_{eff} of connected hidden units (such that $\max_{i,v} |w_{i\mu}(v)| > 0$) against effective strength penalty, for L_1 and L_1^2 penalties. For L_1 penalty $\lambda_1^{eff} = \lambda_1$; for L_1^2, $\lambda_1^{eff} = \lambda_1^2 \frac{1}{NMq} \sum_{\mu,i,v} |w_{\mu i}(v)|$.

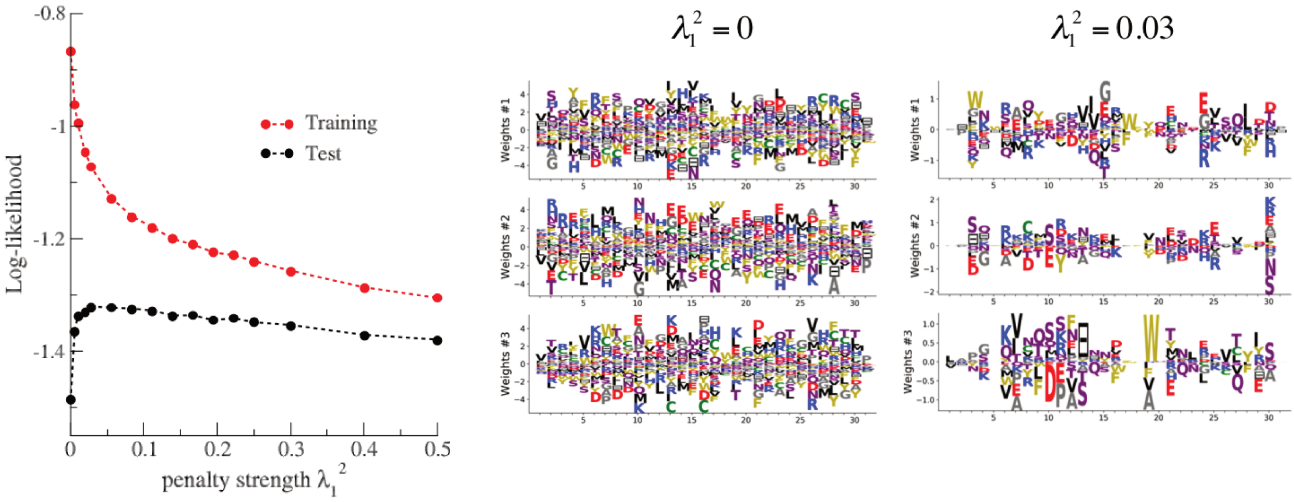

Figure 14.5: Sparsity-performance trade-off for RBM trained on the MSA of WW domain. Left: log-likelihood vs λ_1^2. Right: Selected features at $\lambda_1^2 = 0, 0.03$

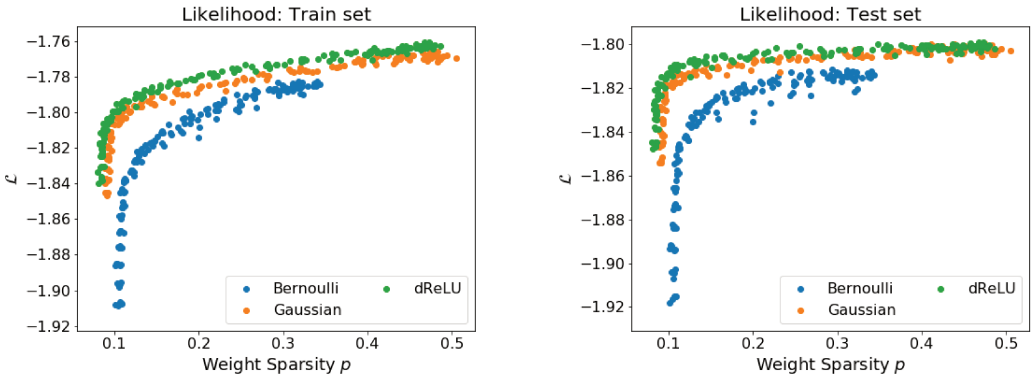

Figure 14.6: Sparsity-performance trade-off on Lattice Proteins. (a),(b) Weight sparsity - Likelihood scatter plots; each point is a RBM trained with a different regularization parameter.

practice, we also evaluate the weight sparsity p. We show the sparsity-likelihood scatter plot in Fig. 14.6 for Lattice Proteins. We observe a plateau in the test likelihood, which shows a large degeneracy of solutions with identical test set performance. If we choose a penalty value such that the model is around the elbow of the curve, we obtain a model having both high likelihood performance and high interpretability.

14.2 TRANSITION TO COMPOSITIONAL PHASE

Besides generative performance, the nature of the representation also changes when varying M and λ_1^2. For LP data sets, we fix $\lambda_1^2 = 0.025$, and vary $M \in [1, 400]$. For very low M, each hidden unit tries to explain as much site covariation as possible, whereas for large M, units focus more on smaller modes such as contacts and the sparsity of the weights decreases, see Fig. 14.8(a). After some point $M \sim 100$, the modes of covariation cannot be decomposed into thinner ones anymore, and duplicate features appear, as can be seen from the distribution of nearest-neighbour overlaps $O_{\mu\nu} = \frac{\sum_{i,v} w_{\mu i}(v) w_{\nu i}(v)}{\sqrt{\left(\sum_{i,v} w_{\mu i}(v)^2\right)(w_{\nu i}(v)^2)}}$ in Fig. 14.8(b). Conversely, the number of simultaneously active hidden units per sequence L grows, with a continuous transition from a ferromagnetic-like phase to a compositional one. Interestingly, once a reweighting is applied to account for overlaps, we find that L does not scale as M, see Fig. 14.8(c). L tends to saturate at about 12 for $M \sim 100$, a number that arise by the data distribution rather than by M. Lastly, Fig. 14.8(d) shows the scaling $L \sim \frac{1}{p}$ is qualitatively correct. We note that though generative performance is best for

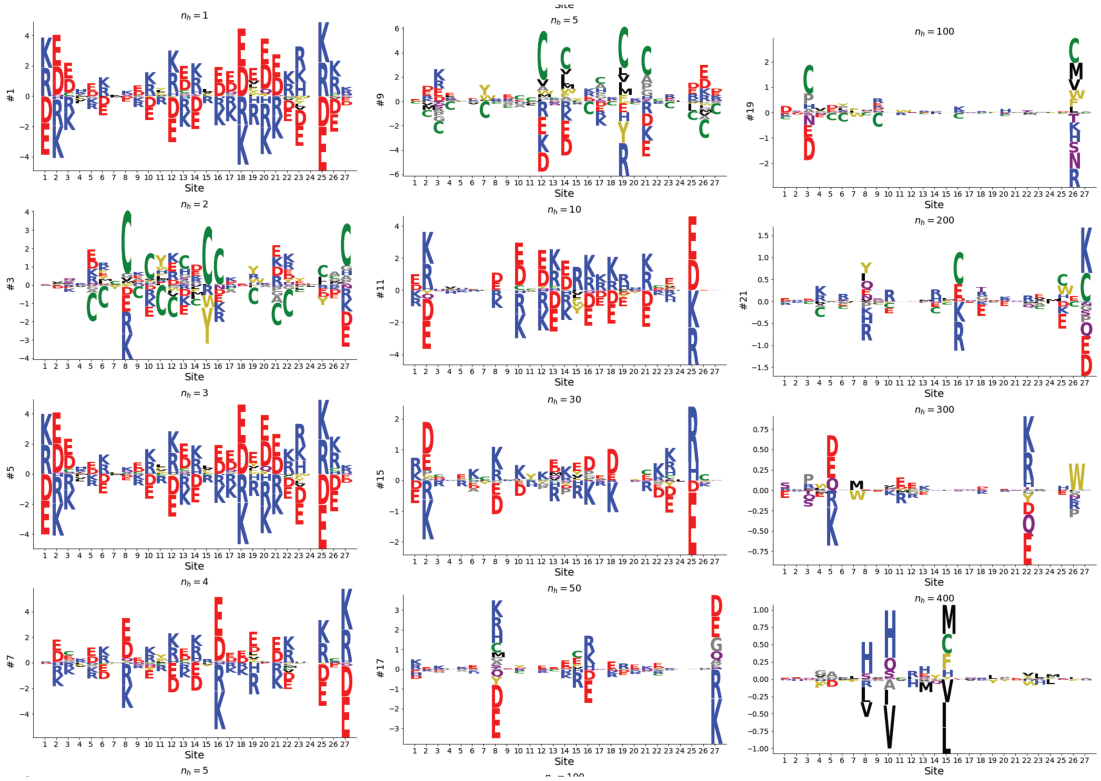

Figure 14.7: Typical weights learnt by RBMs on Lattice Proteins for $M \in [1, 400]$, $\lambda_1^2 = 0.025$. For each RBM, one weight is shown, with sparsity p_μ close to the median sparsity p

very large M, these values are not optimal for interpretation, as the overlaps between hidden units become large. This gives rise to large correlations, such that an evolutionary pattern may be 'split' in two overlapping hidden units. Intermediate values of $M \sim 100$ are therefore optimal for intepretability.

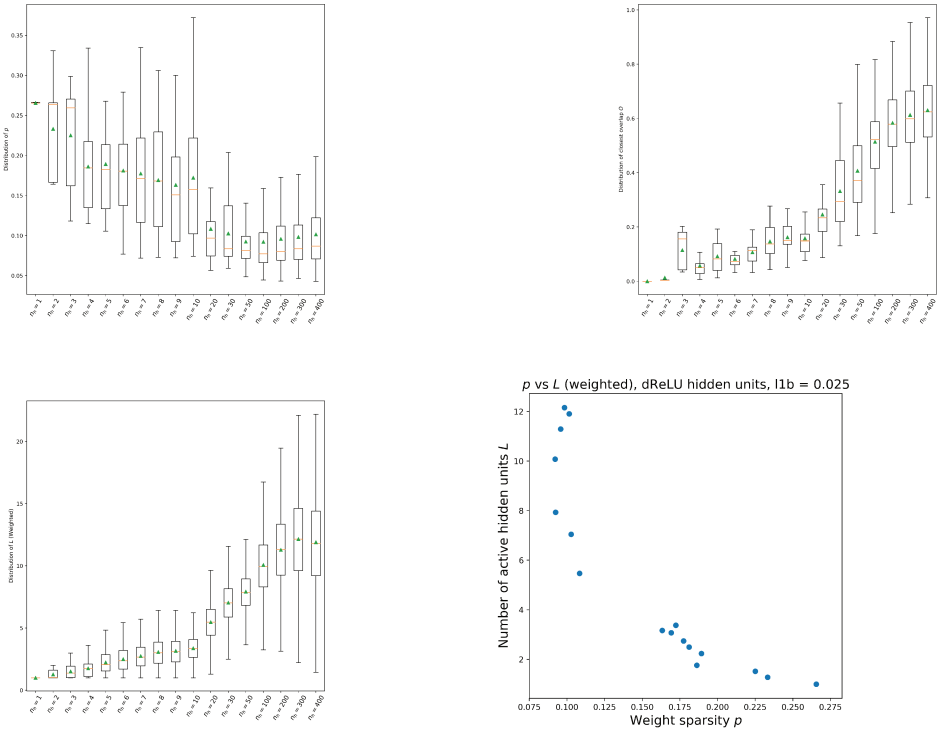

Figure 14.8: Evolution of the representation as function of the number of hidden units. (a) Distribution of weight sparsities p_μ: larger M induce sparser weights (b) Distribution of maximum weight overlaps $O_\mu = \max_{\nu \neq \mu} |O_{\mu\nu}|$: for large M, weights are strongly overlapping. (c) Distribution of the number of simultaneously activated hidden units $L(\mathbf{v})$. We use reweighted participation ratios to correct for overlapping features. (d) Scatter plot p-$\langle L \rangle$

SUMMARY

To summarize, the systematic study suggests that:

- More general potentials like dReLU perform better than the simpler quadratic and Bernoulli potentials;
- L_1^2 regularization is more robust than standard L_1 regularization.
- There exist values of sparsity regularization penalties allowing for both good generative performance and interpretability.
- As the number of hidden units increases, more features are captured and generative performance improve. Moreover, a compositional regime appears, in which a few hidden units are significantly active for each sequence. Beyond some point, increasing M simply adds duplicate hidden units and marginally enhances performance, while making interpretation trickier.

Currently, selecting M and λ_1^2 relies on manual or exhaustive searching. It would be very helpful to find good rationales for specifying these factors a priori, and possibly adjust them throughout training.

DISCUSSION

To summarize, we have shown that RBM could be a promising tool for studying protein coevolution. RBM are capable of extracting a variety of structural, functional and phylogenic information about protein families, with surprising accuracy. To the best of our knowledge, this is unique, compared to other coevolutionary approaches such as DCA, Sectors or Specificity-Determining Positions. The key idea is to enforce RBM to lie in a compositional regime, in which each sequence activates a few hidden units. These hidden units, which may be activated by very different sequences, therefore reflect the underlying function of the sequences rather than their phylogenic origin. Here, we have benchmarked RBM on well-studied protein families, but application to less known protein families could prove very useful for formulating hypothesis before performing experiments. In contrast, traditional approaches are based on knowledge of the protein structure and manual or phylogenic analysis of MSA; they are therefore limited both in the size of data that must be handled and in the complexity of the formulated hypothesis.

Then, RBM combined with conditional and low temperature sampling can be used to design new artificial sequences with predicted function, based on the hidden unit interpretations. In particular, artificial sequences corresponding to unseen combinations of hidden unit activities could have a different function than all of the existing natural sequences. RBM protein design could be used in conjunction with traditional protein design strategies based on physical models of protein folding/docking. For instance, several protein design pipelines begin by computing a position-weight matrix from available natural sequences; they are then used to score sequences first, and keep only sequences with relatively high score before testing them with the physical model [165]. This is basically equivalent to drawing sequences to be tested from an independent model learnt on data. Instead, we could use RBM or conditional RBM for that purpose: they have significantly lower entropy than independent models, such that the size of the sequence space to be tested by costly physical methods could be largely reduced.

Compared to PCA or DCA, an important downside is the requirement to adjust three hyperparameters, namely the hidden unit potential, weight sparsity regularization λ_1^2 and the number of hidden units M. We have provided rationales for this: dReLU hidden units are always better, and M, λ_1^2 are adjusted

so as to achieve both high likelihood and high interpretability. Moreover, even though improvements were introduced for this purpose, training is only approximate and significantly longer, as well as less reproducible as the likelihood is not convex. Therefore, it is important to check the robustness of the conclusions drawn from weight logo by repeating the training with different seed and parameters. Better training algorithms, automated parameter selection and perhaps different regularization schemes would certainly improve the method.

We have briefly investigated on Lattice Proteins whether other feature extraction algorithm, such as ICA, Sparse PCA and Sparse autoencoders - which are all simpler to train - could reproduce the results found here. Though some similarities exist, results were significantly worse in practice, with many false positive contacts, or non biological modes [37]. The takeaway message is that both probabilistic modeling (rather than variance explanation or sequence reconstruction) and interaction-based representations (hidden nodes must encode collective mode rather than single site variability) are crucial for retrieving the results presented here. Moreover, besides RBM, other algorithms that learn both a data representation and a probability distribution were recently developed for this purpose: Variational Autoencoders (VAE) [77] and Generative Adversarial Networks (GAN) [14]. We have ruled out GAN fairly quickly, as i) there is currently no robust method for training GAN on discrete data; for instance text generation is based on Recurrent Neural Networks architectures ii) One cannot compute easily the probability of a configuration using a GAN. Research in GAN is moving forward fairly quickly, and this could of course change in the near future. On the other hand, VAE are suited for our purposes, and were recently applied to protein sequence data for fitness prediction [251, 252]. As RBM, VAE feature high-order interactions and were shown to outperform DCA for fitness prediction in some cases. They can also learn a representation of the sequence space, useful for exploration. Our work differs in several important points: our RBM is an extension of direct-based coupling approaches, requires much less hidden units (about 10 to 50 times less than [251] and [252]), has a simple architecture with two layers carrying sequences and representations, infers interpretable weights with biological relevance, and can be easily tweaked to design sequences with desired statistical properties. In contrast, the low-dimensional representation shown for the β lactamase protein in Fig. 4 of [252] merely reflects phylogenic proximity rather than functional similarity. It is of course not definitive, as one may find a way to emulate a compositional regime using different variants of the VAE presented in this article.

Beyond individual protein families, RBM could be used to find general principles of natural protein design. From one protein family to another, we have noticed several common features, such as stereotyped contacts or functional

loop diversification. It also seems that compositionality could be an ubiquitous feature of protein fitness landscapes, and may be a crucial for evolvability and functional diversification. Other future projects include the development of systematic methods for identifying function-determining sites or intrinsically disordered protein regions. In addition, it would be very interesting to use RBM to determine evolutionary paths between two, or more, protein sequences in the same family, but with distinct phenotypes. In principle, RBM could reveal how functionalities continuously change along the paths, and provide a measure of viability of intermediary sequences. It could also be powerful for estimating evolutionary distances between sequences; this could be used to trace back the evolutionary history of protein families, or detect homologs within a protein family.

Finally, generalization of our approach to other sequential genomic data such as RNA and antibodies is straightforward, and could also lead to interesting developments.

Part V

CONCLUSION

Over the last decade, the statistical physics community has mostly focused on pairwise interaction models for unsupervised data analysis purposes. The rationale behind this choice is multiple. Firstly, pairwise interaction model are justified by a Maximum Entropy principle: it is the *minimal* model that can account for both the mean and correlations of the data. Pairwise interaction are indeed the traditional form of interactions in physics (e.g. gravity, electrostatic, Van Der Waals,...), whereas high-order interactions are often unnecessary to explain large scale collective behaviours of complex systems, and are typically washed away by the renormalization group. Secondly, pairwise interactions models are often very convenient to interpret as causal links between individual units within the system. These causal links can be related to the underlying biological features of the system: synapses in biological neural networks, protein interaction in gene networks, visual attention in bird flocks, contacts in protein sequences, etc. This is in stark contrast with traditional unsupervised models, such as clustering, PCA, or deep networks: these approaches focus mostly on the structure of the data manifold itself rather than on the set of constraint that give rise to it. Therefore there is often no biological/physical interpretation associated to the model parameters or representation inferred for techniques such as PCA or deep networks. Lastly, pairwise models are essentially the only available models ! Indeed, the number of interaction terms involving k units scales as N^k, which is unreasonable for finite data size; it is already doubtful that all pairwise couplings are inferred correctly for a typical data size.

Each of these arguments can be challenged. First, incorporating within the model *all* the second order moments and *none* of the high-order moment is questionable: some second order moments are very noisy (e.g. for rare amino-acids), whereas other high order moments are very strong. Moreover, unlike in physics, high-order interactions are ubiquitous in biological systems: high-order epistasis is systematic in proteins, glial cells mediate high-order interactions in neural networks, and cooperation of more than two proteins are frequent in gene networks; inferred high-order interactions could therefore correspond to biological properties as well. Besides, interactions inferred from second order statistics are often effective in practice, in the sense that they reflect statistical interactions rather than physical ones. Lastly, the combinatorial explosion of the parameter space is not a fatality: tractable high-order models such as ICA and RBM or even mixture models have been around for decades; the key idea is to incorporate only some high-order interactions via a non-linearity in the Hamiltonian.

In this thesis, we have shown that RBM, a high-order interaction model that also learns a representation of data could integrate very well within the toolbox of the statistical physicist. RBM are justified theoretically by the necessity to

use non-linearities to model complex multimodal data, and the emergence of a compositional phase in which a large diversity of attractors is produced from different combinations of features. Crucially, our work provides a conceptual framework for interpreting the representation and the model parameters: provided that the weights are sparse, they can be interpreted as typical parts of data configurations. Clearly, there is a wide conceptual and technical gap between theoretical understanding of RBM and of deep networks, but our work may be a small step toward the right direction.

Secondly, we have overcome several weaknesses of standard RBM parametrisation and training, to the point where they can compete with state-of-the-art methods such as Variational Autoencoders or Generative Adversarial Networks on relatively small data sets such as MNIST or proteins. This work suggests that RBM can achieve a good compromise between model interpretability and generative performance.

Using RBM, we were able to infer a wealth of functional, structural and phylogenic information from protein sequence data only. Moreover, RBM can recombine natural sequences into new artificial ones, that are putatively functional and may have different functionalities than natural sequences. RBM may find applications for both designing artificial proteins and elucidating general principles underlying natural protein design. In the future, RBM may be used to retrace the evolutionary history of protein families.

We do not expect Restricted Boltzmann Machines to replace Boltzmann Machines, as the later remain best suited for inferring interaction networks such as contacts. However, RBM are significantly better at elucidating collective modes of variation of data, and may find applications for this purpose in other important domains such as neuroscience, RNA analysis and gene networks.

Part VI

APPENDIX

A

ANNEX: TECHNICAL DETAILS OF TRAINING ALGORITHM

A.1 ADDITIONAL INFORMATIONS FOR SGD

RBM are initialized as follows:

- The visible layer fields as, $g_i(v) = \log f_i(v) - \frac{1}{q_v}\sum_{v'=1}^{q} \log f_i(v')$, *i.e.* the fields of the independent model.
- The parameters of the hidden unit potential as $g_\mu = 0$ if Bernoulli; $\gamma_\mu = 1, \theta_\mu = 0 \ \ \forall \mu$ if Gaussian or ReLU, and $\gamma_\mu^+ = \gamma_\mu^- = 1$, $\theta_\mu^+ = \theta_\mu^- = 0$ for dReLU.
- The weights as random, independent Gaussian $w_{i\mu}(a) \sim \sqrt{\frac{0.01}{N}}\mathcal{N}(0,1)$. Indeed, $\mathbf{w} = 0$ is a critical point with a vanishing gradient, hence the initial weights should not be zero; the normalization choice ensures that the initial inputs I_μ are of order 1.

The learning rate is set to lr_i for the d_a% first updates, after which it decays in a geometric fashion until it reaches lr_f. The number of epochs depends on the numerical experiments and was handpicked; a good rule of thumb is that the lower the final entropy of P (e.g. stronger correlations, more hidden units, less regularization), the longer the training should be. The data set is shuffled after each epoch.

A.2 CHOICE OF INITIAL POTENTIALS FOR PT/ APT

Both Parallel Tempering and Augmented Parallel Tempering require choosing an initial independent distribution $P_0(\mathbf{v},\mathbf{h}) = \frac{1}{Z_0}e^{-E_0(\mathbf{v},\mathbf{h})}$ with energy $E_0 = -\sum_i \mathcal{U}_i^0(v_i) - \sum_\mu \mathcal{U}_\mu^0(h_\mu)$. We choose formally P_0 so as to minimize $D_{KL}(P_d|P_0)$. Since the probability factorizes, $D_{KL}(P_d|P_0) = \sum_i D_{KL}\left(P_d(v_i)|P_0(v_i)\right) + \sum_\mu D_{KL}\left(P_d(h_\mu)|P_0(h_\mu)\right)$ is a sum of individual term which can be optimized independently. Here, $P_d(h_\mu) = \int P(h_\mu|\mathbf{v})P_d(\mathbf{v})$.

For categorical visible variables we obtain the standard independent model fields:

$$g_i^0(a) = \log \langle \delta_{v_i,a} \rangle_d - \frac{1}{q_v} \sum_b \log \langle \delta_{v_i,b} \rangle_d \tag{A.1}$$

For Gaussian hidden units, we obtain:

$$\begin{aligned} \theta_\mu^0 &= 0 \\ \gamma_\mu^0 &= 1 \end{aligned} \tag{A.2}$$

At any time, since the hidden units are normalized. For Bernoulli and dReLU hidden units, since their statistics evolve during training, the corresponding hidden potential parameters must be adjusted dynamically by gradient descent. The updates write:

$$g_\mu^0 \to g_\mu^0 + \mathrm{lr} \left(\left\langle \Gamma'_\mu(I_\mu(\mathbf{v}) \right\rangle_d - \Gamma'^0_\mu(0) \right) \tag{A.3}$$

And:

$$\begin{aligned} \gamma_\mu^{+0} &\to \gamma_\mu^{+0} + \mathrm{lr} \left[\left\langle \partial_{\gamma^+} \Gamma_\mu(I_\mu(\mathbf{v}) \right\rangle_d - \partial_{\gamma^+} \Gamma_\mu^0(0) \right] \\ \gamma_\mu^{-0} &\to \gamma_\mu^{-0} + \mathrm{lr} \left[\left\langle \partial_{\gamma^-} \Gamma_\mu(I_\mu(\mathbf{v}) \right\rangle_d - \partial_{\gamma^-} \Gamma_\mu^0(0) \right] \\ \theta_\mu^{+0} &\to \theta_\mu^{+0} + \mathrm{lr} \left[\left\langle \partial_{\theta^+} \Gamma_\mu(I_\mu(\mathbf{v}) \right\rangle_d - \partial_{\theta^+} \Gamma_\mu^0(0) \right] \\ \theta_\mu^{-0} &\to \theta_\mu^{-0} + \mathrm{lr} \left[\left\langle \partial_{\theta^-} \Gamma_\mu(I_\mu(\mathbf{v}) \right\rangle_d - \partial_{\theta^-} \Gamma_\mu^0(0) \right] \end{aligned} \tag{A.4}$$

Where Γ^0 is the c.g.f. evaluated at the initial parameters. We use the same learning rate as for the gradient descent.

A.3 IMPLEMENTATION OF THE REPARAMETRIZATION TRICK FOR BERNOULLI AND DRELU

A.3.1 *Bernoulli*

Here, we recall the reparametrization used for Bernoulli potentials, which is equivalent to the centering trick.

$$\begin{aligned} U_\mu(h) &= -g_\mu h \\ g_\mu &= \tilde{g_\mu} - \left\langle I_\mu(\mathbf{v}) \right\rangle_{\mathbf{v}\sim P_d} \end{aligned} \tag{A.5}$$

This choice ensures that the input is always centered, without limiting the capacity of the model. The partial derivatives, cross-derivative and final gradient equations are:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \tilde{g_\mu}} = \frac{\partial \mathcal{L}}{\partial g_\mu} &= = \left\langle h_\mu \right\rangle_{\mathbf{v}\sim P_d} \\ \frac{\partial \delta_\mu}{\partial w_{i\mu(a)}} &= -\left\langle \delta_{v_i,a} \right\rangle_{\mathbf{v}\sim P_d} \\ \frac{\partial \mathcal{L}}{\partial w_{i\mu}(a)} &= \left\{ \left\langle \delta_{v_i,a} h_\mu \right\rangle_{\mathbf{v}\sim P_d} - \left\langle \delta_{v_i,a} \right\rangle_{\mathbf{v}\sim P_d} \left\langle h_\mu \right\rangle_{\mathbf{v}\sim P_d} \right\} \\ &\quad - \left\{ \left\langle \delta_{v_i,a} h_\mu \right\rangle_{\mathbf{v}\sim P} - \left\langle \delta_{v_i,a} \right\rangle_{\mathbf{v}\sim P_d} \left\langle h_\mu \right\rangle_{\mathbf{v}\sim P} \right\} \end{aligned} \tag{A.6}$$

A.3.2 *dReLU*

We start by introducing the following change of variable for the potential parameters:

$$\begin{cases} \gamma^+ = \frac{\gamma}{1+\eta} \\ \gamma^- = \frac{\gamma}{1-\eta} \\ \theta^+ = \theta + \frac{\Delta}{1+\eta} \\ \theta^- = \theta - \frac{\Delta}{1-\eta} \end{cases} \tag{A.7}$$

Or equivalently:

$$\begin{cases} \gamma &= \frac{2\gamma^+\gamma^-}{\gamma^+ + \gamma^-} \\ \eta &= \frac{\gamma^- - \gamma^+}{\gamma^- + \gamma^+} \\ \theta &= \frac{\gamma^-}{\gamma^+ + \gamma^-}\theta^+ + \frac{\gamma^+}{\gamma^+ + \gamma^-}\theta^- \\ \Delta &= \frac{2\gamma^+\gamma^-}{(\gamma^+ + \gamma^-)^2}(\theta^+ - \theta^-) \end{cases} \tag{A.8}$$

This parametrization helps better quantifying and interpreting the non-gaussianity of the potential. γ and θ are the same parameters as for the quadratic potential, they control the curvature (resp. offset) of the potential, *i.e.* the slope and the offset of the transfer function. $\eta \in [-1, 1]$ quantifies the asymmetry of the potential: for $\eta = \pm 1$, $\mathcal{U}(h) = \infty \ \ \forall x \lesseqgtr 0$ and the hidden unit becomes a single ReLU. Δ quantifies the first derivative jump: for $\Delta < 0$, the potential has two local minima and the distribution is bimodal whereas for $\Delta > 0$, there is a single minimum with singular curvature, and the distribution is sparse. In terms of moments, θ, γ, η and Δ control respectively the mean, variance, skewness and kurtosis of the distribution of h_μ. If $\Delta = \eta = 0$, the potential is effectively quadratic.

The cumulant generating function and its moments rewrite:

- $\Gamma(I) = \log\left[\Phi\left(\frac{\frac{\Delta}{\sqrt{1+\eta}} - \sqrt{1+\eta}(I-\theta)}{\sqrt{\gamma}}\right)\sqrt{\frac{1+\eta}{\gamma}} + \Phi\left(\frac{\frac{\Delta}{\sqrt{1-\eta}} + \sqrt{1-\eta}(I-\theta)}{\sqrt{\gamma}}\right)\sqrt{\frac{1-\eta}{\gamma}}\right]$
- If $\Delta > 0$:

  $$H(I) = \text{ReLU}\left(\frac{(1+\eta)(I-\theta) - \Delta)}{\gamma}\right) - \text{ReLU}\left(\frac{(1-\eta)(I-\theta) + \Delta)}{\gamma}\right)$$

  If $\Delta < 0$:

  $$H(I) = \begin{cases} \frac{\sqrt{1+\eta}(I-\theta) - \frac{\Delta}{\sqrt{1+\eta}}}{\gamma} & \text{if} \quad I \geq \theta + \frac{2\eta}{(\sqrt{1+\eta}+\sqrt{1-\eta})^2}\Delta \\ \frac{\sqrt{1-\eta}(I-\theta) + \frac{\Delta}{\sqrt{1-\eta}}}{\gamma} & \text{if} \quad I \leq \theta + \frac{2\eta}{(\sqrt{1+\eta}+\sqrt{1-\eta})^2}\Delta \end{cases}$$

- $P[h > 0|I] \equiv p^+ = 1 - p^- = \frac{\Phi\left(\frac{\frac{\Delta}{\sqrt{1+\eta}} - \sqrt{1+\eta}(I-\theta)}{\sqrt{\gamma}}\right)\sqrt{1+\eta}}{\Phi\left(\frac{\frac{\Delta}{\sqrt{1+\eta}} - \sqrt{1+\eta}(I-\theta)}{\sqrt{\gamma}}\right)\sqrt{1+\eta} + \Phi\left(\frac{\frac{\Delta}{\sqrt{1-\eta}} + \sqrt{1-\eta}(I-b)}{\sqrt{\gamma}}\right)\sqrt{1-\eta}}$

- $$\langle h|I\rangle = \frac{1}{\gamma}\left\{(I-\theta)(1+\eta(p_+-p_-)) - \Delta(p_+-p_-) + \right.$$
$$\left. \frac{2\eta\sqrt{\gamma}}{\sqrt{1+\eta}\Phi\left(\frac{\frac{\Delta}{\sqrt{1+\eta}}-\sqrt{1+\eta}(I-\theta)}{\sqrt{\gamma}}\right) + \sqrt{1-\eta}\Phi\left(\frac{\frac{\Delta}{\sqrt{1-\eta}}+\sqrt{1-\eta}(I-\theta)}{\sqrt{\gamma}}\right)}\right\}$$
$$\equiv \frac{1}{a}\mathcal{E}(I-\theta,\Delta,\gamma,\eta)$$

- $$\mathrm{Var}[h|I] = \frac{1}{\gamma}\left\{1+\eta(p_+-p_-)\right.$$
$$+p_+p_-\left[2\frac{\Delta}{\sqrt{\gamma}} - 2\eta\frac{I-\theta}{\sqrt{\gamma}}\right]\left[\frac{2\Delta}{\sqrt{\gamma}} - 2\eta\frac{I-\theta}{\sqrt{\gamma}} - \frac{\sqrt{1+\eta}}{\Phi\left(\frac{\frac{\Delta}{\sqrt{1+\eta}}-\sqrt{1+\eta}(I-\theta)}{\sqrt{\gamma}}\right)} - \frac{\sqrt{1-\eta}}{\Phi\left(\frac{\frac{\Delta}{\sqrt{1-\eta}}+\sqrt{1+\eta}}{\sqrt{\gamma}}\right.}\right.$$
$$\left.-\frac{2\eta}{\sqrt{\gamma}\left(\sqrt{1+\eta}\Phi\left(\frac{\frac{\Delta}{\sqrt{1+\eta}}-\sqrt{1+\eta}(I-\theta)}{\sqrt{\gamma}}\right) + \sqrt{1-\eta}\Phi\left(\frac{\frac{\Delta}{\sqrt{1-\eta}}+\sqrt{1-\eta}(I-\theta)}{\sqrt{\gamma}}\right)\right)}\mathcal{E}(I-\theta,\Delta,\gamma,\eta\right.$$
$$\equiv \frac{1}{\gamma}(1+\mathcal{V}(I-\theta,\Delta,\gamma,\eta))$$

And the derivatives of Γ are, by application of the chain rule:

- $\partial_\theta\Gamma(I) = \langle h|I\rangle$
- $\partial_\gamma\Gamma(I) = -\frac{1}{2}\left\langle \frac{1}{1+\eta}\max(h,0)^2 - \frac{1}{1-\eta}\min(h,0)^2|I\right\rangle$
- $\partial_\Delta\Gamma(I) = -\left\langle \frac{1}{1+\eta}\max(h,0) - \frac{1}{1-\eta}\min(h,0)|I\right\rangle$
- $\partial_\eta\Gamma(I) = \left\langle \frac{\gamma}{2}\left(\frac{1}{(1+\eta)^2}\max(h,0)^2 - \frac{1}{(1-\eta)^2}\min(h,0)^2\right) + \Delta\left(\frac{1}{(1+\eta)^2}\max(h,0) - \frac{1}{(1-\eta)^2}\min\right.\right.$

With this parameterization, the following transformation: $\gamma \to \lambda^2\gamma$, $I \to \lambda I$, $\theta \to \lambda\theta$, $\Delta \to \lambda\Delta$ leaves the effective potential invariant (up to an additive term), hence γ can be chosen arbitrarily in the model. On the other hand, due to the

presence of order 3 terms in Γ, changing θ cannot be compensated by changing the visible layer fields like in the Gaussian case. Rather, we proceed as in the centering trick, and set:

$$\theta = \tilde{\theta} + \langle I \rangle \tag{A.9}$$

And choose γ such that:

$$\mathrm{Var}[h]_{MSA} = 1 \tag{A.10}$$

At this point, one should realize that Eqn. (A.10) cannot be solved analytically. It is extremely tempting to not perform exact batch normalization, and use the same formula for γ as in the Gaussian case; after all, we could expect that the distribution would be approximately normalized and that divergence problems are still solved even if exact normalization is not achieved. It matters however when the RBM is regularized (e.g., with L_1 norm on the weights), as the outcome of regularized training depends on the parametrization choice. In fact, when trying to train regularized dReLU with the Gaussian gauge choice, no optimum was found, but instead an asymptotic divergence of the form $\mathbf{w} \to 0, \Delta \to -\infty$ appeared, such that the inputs of the hidden unit go to zero but the slope of the average activity goes to infinity. Thus exact batch normalization is required, and we must proceed. We rewrite Eqn. (A.10) as:

$$\begin{aligned}
1 &= \langle \mathrm{Var}[h|\mathbf{v}] \rangle_{MSA} + \mathrm{Var}[\langle h|\mathbf{v} \rangle]_{MSA} \\
\iff 1 &= \frac{1}{\gamma^2} \mathrm{Var}[\mathcal{E}(I(\mathbf{v}) - \theta, \Delta, \gamma, \eta)]_{MSA} + \frac{1}{\gamma} \left(1 + \langle \mathcal{V}(I(\mathbf{v}) - \theta, \Delta, \gamma, \eta) \rangle_{MSA}\right) \\
\iff \gamma &= \frac{1}{2} \Big\{ 1 + \langle \mathcal{V}(I(\mathbf{v}) - \theta, \Delta, \gamma, \eta) \rangle_d \\
&\quad + \sqrt{(1 + \langle \mathcal{V}(I(\mathbf{v}) - \theta, \Delta, \gamma, \eta) \rangle_d)^2 + 4\mathrm{Var}[\mathcal{E}(I(\mathbf{v}) - \theta, \Delta, \gamma, \eta)]_d} \Big\} \\
&\quad \equiv G(\gamma, \theta, \Delta, \eta, P_d)
\end{aligned} \tag{A.11}$$

The above implicit equation A.11 is solved iteratively through $\gamma^{(t+1)} = \Gamma(\gamma^{(t)}, \delta, \theta, \eta, P_d)$. As for the Gaussian case, we evaluate the expectation and variances on a mini-batch before computing the gradient, and we perform only one iteration step per gradient update. Furthermore, we use an exponential smoothing $\gamma^{(t+1)} = \rho\Gamma(\gamma^{(t)}, \delta, \theta, \eta, P_d) + (1 - \rho)\gamma^{(t)}$ after a while, with $\rho_i = 1$

and $\rho \to 0$ to ensure convergence. Lastly, unlike the Gaussian case, the non-linear moments estimators can have a very large variance, particularly when hidden unit h_μ encodes for a very rare feature; in that case, the variance can decrease abruptly, yielding large fluctuations of γ. To alleviate this problem, we bound $\gamma^{(t+1)} \geq \frac{3}{4}\gamma^{(t)}$. To obtain the gradient, one needs to compute the derivatives of $\mathcal{E}, \mathcal{V}$. They can be obtained, in principle, by automatic symbolic differentiation; however, numerical stability problems arose with the expression obtained. Instead we derived analytically the derivatives as follows. When they appear, ξ,ξ' denote any of the I,θ,Δ,η.

1. $I_+ = \frac{\Delta}{\sqrt{(1+\eta)\gamma}} - \sqrt{1+\eta}\frac{(I-\theta)}{\sqrt{\gamma}}$
2. $I_- = \frac{\Delta}{\sqrt{(1-\eta)\gamma}} + \sqrt{1-\eta}\frac{(I-\theta)}{\sqrt{\gamma}}$
3. $\phi_+ = \phi(I_+)$
4. $\phi_- = \phi(I_-)$
5. $Z = \phi_+\sqrt{1+\eta} + \phi_-\sqrt{1-\eta}$
6. $p_+ = 1 - p_- = \frac{\phi_+\sqrt{1+\eta}}{Z}$
7. $\mathcal{E} = (I-\theta)(1+\eta(p_+ - p_-)) - \Delta(p_+ - p_-) + \frac{2\eta\sqrt{\gamma}}{Z}$
8. $\mathcal{V} = \eta(p_+ - p_-) + p_+p_-\left[2\frac{\Delta}{\sqrt{\gamma}} - 2\eta\frac{I-\theta}{\sqrt{a}}\right]\left[\frac{2\Delta}{\sqrt{\gamma}} - 2\eta\frac{I-\theta}{\sqrt{\gamma}} - \frac{\sqrt{1+\eta}}{\phi_+} - \frac{\sqrt{1-\eta}}{\phi_-}\right] - \frac{2\eta}{\sqrt{\gamma}Z}$
9. $\frac{\partial I_+}{\partial\gamma} = -\frac{1}{2\gamma}I_+$
10. $\frac{\partial I_-}{\partial\gamma} = -\frac{1}{2\gamma}I_-$
11. $\frac{\partial I_+}{\partial I} = -\frac{\sqrt{1+\eta}}{\sqrt{\gamma}}$
12. $\frac{\partial I_-}{\partial I} = \frac{\sqrt{1-\eta}}{\sqrt{\gamma}}$
13. $\frac{\partial I_+}{\partial\theta} = \frac{\sqrt{1+\eta}}{\sqrt{\gamma}}$
14. $\frac{\partial I_-}{\partial\theta} = -\frac{\sqrt{1-\eta}}{\sqrt{\gamma}}$

15. $\frac{\partial I_+}{\partial \Delta} = \frac{1}{\sqrt{\gamma(1+\eta)}}$

16. $\frac{\partial I_-}{\partial \Delta} = \frac{1}{\sqrt{\gamma(1-\eta)}}$

17. $\frac{\partial I_+}{\partial \eta} = \frac{-1}{2\sqrt{\gamma(1+\eta)}}(I - \theta + \frac{\Delta}{1+\eta})$

18. $\frac{\partial I_-}{\partial \eta} = \frac{-1}{2\sqrt{\gamma(1-\eta)}}(I - \theta - \frac{\Delta}{1-\eta})$

19. $\frac{\partial^2 I_+}{\partial \gamma \partial I} = \frac{1}{2}\sqrt{\frac{1+\eta}{\gamma^3}}$

20. $\frac{\partial^2 I_-}{\partial \gamma \partial I} = -\frac{1}{2}\sqrt{\frac{1-\eta}{\gamma^3}}$

21. $\frac{\partial^2 I_+}{\partial \Delta \partial I} = 0$

22. $\frac{\partial^2 I_-}{\partial \Delta \partial I} = 0$

23. $\frac{\partial^2 I_+}{\partial \eta \partial I} = -\frac{1}{2}\sqrt{\frac{1}{(1+\eta)\gamma}}$

24. $\frac{\partial^2 I_-}{\partial \eta \partial I} = -\frac{1}{2}\sqrt{\frac{1}{(1-\eta)\gamma}}$

25. $\frac{\partial \phi_+}{\partial I_+} = I_+\phi_+ - 1$

26. $\frac{\partial \phi_-}{\partial I_-} = I_-\phi_- - 1$

27. $\frac{\partial p_+}{\partial I_+} = p_+p_-(I_+ - \frac{1}{\phi_+})$

28. $\frac{\partial p_+}{\partial I_-} = -p_+p_-(I_- - \frac{1}{\phi_-})$

29. $\frac{\partial p_+}{\partial \xi} = p_+p_-\left\{(I_+ - \frac{1}{\phi_+})\frac{\partial I_+}{\partial \xi} - (I_- - \frac{1}{\phi_-})\frac{\partial I_-}{\partial \xi}\right\}$ for $\xi \in \{\gamma, \theta, \Delta, \eta, I\}$

30. $\frac{\partial^2 p_+}{\partial \xi \xi'} = -(p_+ - p_-)p_+p_-\left\{(I_+ - \frac{1}{\phi_+})\frac{\partial I_+}{\partial \xi} - (I_- - \frac{1}{\phi_-})\frac{\partial I_-}{\partial \xi}\right\}\left\{(I_+ - \frac{1}{\phi_+})\frac{\partial I_+}{\partial \xi'} - (I_- - \frac{1}{\phi_-})\frac{\partial I_-}{\partial \xi'}\right\} +$
$p_+p_-\left\{(I_+ - \frac{1}{\phi_+})\frac{\partial^2 I_+}{\partial \xi \xi'} - (I_- - \frac{1}{\phi_-})\frac{\partial^2 I_-}{\partial \xi \xi'}\right\} + p_+p_-\left\{\frac{\partial I_+}{\partial \xi}\frac{\partial I_+}{\partial \xi'}\left[1 + \frac{I_+ - \phi_+^{-1}}{\phi_+}\right] - \frac{\partial I_-}{\partial \xi}\frac{\partial I_-}{\partial \xi'}\left[1 + \frac{I_- - \phi_-^{-1}}{\phi_-}\right]\right\}$

31. $\frac{\partial \log Z}{\partial \xi} = p_+(I_+ - \frac{1}{\phi_+})\frac{\partial I_+}{\partial \xi} + p_-(I_- - \frac{1}{\phi_-})\frac{\partial I_0}{\partial \xi}$

32. $\frac{\partial \mathcal{E}}{\partial I} = 1 + \mathcal{V}$

33. $\frac{\partial \mathcal{E}}{\partial \theta} = -\frac{\partial \mathcal{E}}{\partial I}$

34. $\frac{\partial \mathcal{E}}{\partial w_i(a)} = \theta_{v_i,a} \frac{\partial \mathcal{E}}{\partial I}$

35. $\frac{\partial \mathcal{E}}{\partial \gamma} = 2\left[(I-\theta)\eta - \Delta\right] \frac{\partial p_+}{\partial \gamma} + \frac{\eta}{Z\sqrt{\gamma}} - \frac{2\eta\sqrt{\gamma}}{Z} \frac{\partial \log Z}{\partial \xi}$

36. $\frac{\partial \mathcal{E}}{\partial \Delta} = -(p_+ - p_-) + 2\left[(I-\theta)\eta - \Delta\right] \frac{\partial p_+}{\partial \Delta} - \frac{2\eta\sqrt{\gamma}}{Z}$

37. $\frac{\partial \mathcal{E}}{\partial \eta} = (I-\theta)(p_+ - p_-) + 2\left[(I-\theta)\eta - \Delta\right] \frac{\partial p_+}{\partial \eta} + \frac{2\sqrt{\gamma}}{Z} - \frac{2\eta\sqrt{\gamma}}{Z} \frac{\partial \log Z}{\partial \eta}$

38. $\frac{\partial \mathcal{V}}{\partial I} = 4\eta \frac{\partial p_+}{\partial I} + 2\left[(I-\theta)\eta - \Delta\right] \frac{\partial^2 p_+}{\partial I^2} - 2\frac{\eta}{\sqrt{\gamma}Z}\left(\frac{\partial \mathcal{E}}{\partial I} - \mathcal{E}\frac{\partial \log Z}{\partial I}\right)$

39. $\frac{\partial \mathcal{V}}{\partial \theta} = -\frac{\partial \mathcal{V}}{\partial I}$

40. $\frac{\partial \mathcal{V}}{\partial w_i(a)} = \theta_{v_i,a} \frac{\partial \mathcal{V}}{\partial I}$

41. $\frac{\partial \mathcal{V}}{\partial \gamma} = 2\eta \frac{\partial p_+}{\partial \gamma} + 2\left[(I-\theta)\eta - \Delta\right] \frac{\partial^2 p_+}{\partial \gamma \partial I} + 2\frac{\eta}{\sqrt{\gamma}Z}\left(\frac{\mathcal{E}}{2\gamma} + \mathcal{E}\frac{\partial \log Z}{\partial \gamma} - \frac{\partial \mathcal{E}}{\partial \gamma}\right)$

42. $\frac{\partial \mathcal{V}}{\partial \Delta} = 2\eta \frac{\partial p_+}{\partial \Delta} - 2\frac{\partial p_+}{\partial I} + 2\left[(I-\theta)\eta - \Delta\right] \frac{\partial^2 p_+}{\partial \Delta \partial I} + 2\frac{\eta}{\sqrt{\gamma}Z}\left(\mathcal{E}\frac{\partial \log Z}{\partial \Delta} - \frac{\partial \mathcal{E}}{\partial \Delta}\right)$

43. $\frac{\partial \mathcal{V}}{\partial \eta} = p_+ - p_- + 2\eta \frac{\partial p_+}{\partial \eta} + 2(I-\theta)\frac{\partial p_+}{\partial I} + 2\left[(I-\theta)\eta - \Delta\right] \frac{\partial^2 p_+}{\partial \eta \partial I} - \frac{2}{\sqrt{\gamma}Z}\left(\mathcal{E} - \eta\mathcal{E}\frac{\partial \log Z}{\partial \eta} + \eta\frac{\partial \mathcal{E}}{\partial \eta}\right)$

With the above formula, the cross-derivatives and the gradients can be computed. Note again that for sparse features, γ can fluctuate a lot and the above cross derivatives can be very large; we therefore threshold the final gradient for numerical stability.

BIBLIOGRAPHY

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[2] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep learning, vol. 1. MIT press Cambridge, 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, pp. 1097–1105, 2012.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal processing magazine, vol. 29, no. 6, pp. 82–97, 2012.

[7] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, pp. 6645–6649, IEEE, 2013.

[8] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in Proceedings of the 25th international conference on Machine learning, pp. 160–167, ACM, 2008.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in neural information processing systems, pp. 3111–3119, 2013.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, pp. 3104–3112, 2014.

[12] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato, "Phrase-based & neural unsupervised machine translation," arXiv preprint arXiv:1804.07755, 2018.

[13] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," Nature, vol. 542, no. 7639, p. 115, 2017.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, pp. 2672–2680, 2014.

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.

[16] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in Artificial Intelligence and Statistics, pp. 192–204, 2015.

[17] S. Mallat, "Understanding deep convolutional networks," Phil. Trans. R. Soc. A, vol. 374, no. 2065, p. 20150203, 2016.

[18] J. Kadmon and H. Sompolinsky, "Optimal architectures in a solvable model of deep networks," in Advances in Neural Information Processing Systems, pp. 4781–4789, 2016.

[19] M. Baity-Jesi, L. Sagun, M. Geiger, S. Spigler, G. B. Arous, C. Cammarota, Y. LeCun, M. Wyart, and G. Biroli, "Comparing dynamics: Deep neural networks versus glassy systems," arXiv preprint arXiv:1803.06969, 2018.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International conference on machine learning, pp. 448–456, 2015.

[22] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in Advances in Neural Information Processing Systems, pp. 3981–3989, 2016.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826, 2016.

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," science, vol. 220, no. 4598, pp. 671–680, 1983.

[25] E. Gardner, "Maximum storage capacity in neural networks," EPL (Europhysics Letters), vol. 4, no. 4, p. 481, 1987.

[26] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Storing infinite numbers of patterns in a spin-glass model of neural networks," Physical Review Letters, vol. 55, no. 14, p. 1530, 1985.

[27] H. Seung, H. Sompolinsky, and N. Tishby, "Statistical mechanics of learning from examples," Physical review A, vol. 45, no. 8, p. 6056, 1992.

[28] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, "Determining computational complexity from characteristic 'phase transitions'," Nature, vol. 400, no. 6740, p. 133, 1999.

[29] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems," Science, vol. 297, no. 5582, pp. 812–815, 2002.

[30] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, "Spectral redemption in clustering sparse networks," Proceedings of the National Academy of Sciences, vol. 110, no. 52, pp. 20935–20940, 2013.

[31] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Statistical-physics-based reconstruction in compressed sensing," Physical Review X, vol. 2, no. 2, p. 021005, 2012.

[32] H. C. Nguyen, R. Zecchina, and J. Berg, "Inverse statistical problems: from the inverse ising problem to data science," Advances in Physics, vol. 66, no. 3, pp. 197–261, 2017.

[33] S. Cocco, R. Monasson, L. Posani, S. Rosay, and J. Tubiana, "Statistical physics and representations in real and artificial neural networks," Physica A: Statistical Mechanics and its Applications, vol. 504, pp. 45–76, 2018.

[34] J. Tubiana and R. Monasson, "Efficient sampling and parametrization improve restricted boltzmann machines," 2018.

[35] J. Tubiana and R. Monasson, "Emergence of compositional representations in restricted boltzmann machines," Physical review letters, vol. 118, no. 13, p. 138301, 2017.

[36] J. Tubiana, S. Cocco, and R. Monasson, "Learning protein constitutive motifs from sequence data," arXiv preprint arXiv:1803.08718, 2018.

[37] J. Tubiana, S. Cocco, and R. Monasson, "Learning lattice proteins with restricted boltzmann machines : compositional regime and comparative analysis," 2018.

[38] Y. LeCun, "L'apprentissage profond." Lectures at Collège de France, 2016.

[39] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[40] S. MuellerStein, A. Loccisano, S. Firestine, and J. Evanseck, "Principal components analysis: A review of its application on molecular dynamics data," Annual Reports in Computational Chemistry, vol. 2, pp. 233–66, 2006.

[41] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on, pp. 586–591, IEEE, 1991.

[42] D. L. Ruderman and W. Bialek, "Statistics of natural images: Scaling in the woods," in Advances in neural information processing systems, pp. 551–558, 1994.

[43] A. Hyvärinen, J. Karhunen, and E. Oja, Independent component analysis, vol. 46. John Wiley & Sons, 2004.

[44] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," Nature, vol. 381, no. 6583, p. 607, 1996.

[45] A. Ng, "Sparse autoencoder," CS294A Lecture notes, vol. 72, no. 2011, pp. 1–19, 2011.

[46] T.-W. Lee, M. Girolami, A. J. Bell, and T. J. Sejnowski, "A unifying information-theoretic framework for independent component analysis," Computers & Mathematics with Applications, vol. 39, no. 11, pp. 1–21, 2000.

[47] D. L. Ringach, R. M. Shapley, and M. J. Hawken, "Orientation selectivity in macaque v1: diversity and laminar dependence," Journal of Neuroscience, vol. 22, no. 13, pp. 5639–5651, 2002.

[48] J. Zylberberg, J. T. Murphy, and M. R. DeWeese, "A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields," PLoS computational biology, vol. 7, no. 10, p. e1002250, 2011.

[49] T. Poggio and F. Anselmi, Visual cortex and deep networks: learning invariant representations. MIT Press, Cambridge, 2016.

[50] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122, 2013.

[51] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," IEEE Transactions on image processing, vol. 17, no. 1, pp. 53–69, 2008.

[52] D. Taubman and M. Marcellin, JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice, vol. 642. Springer Science & Business Media, 2012.

[53] A. J. Bell and T. J. Sejnowski, "The "independent components" of natural scenes are edge filters," Vision research, vol. 37, no. 23, pp. 3327–3338, 1997.

[54] S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski, "Independent component analysis of electroencephalographic data," in Advances in neural information processing systems, pp. 145–151, 1996.

[55] M. J. McKeown, S. Makeig, G. G. Brown, T.-P. Jung, S. S. Kindermann, A. J. Bell, and T. J. Sejnowski, "Analysis of fmri data by blind separation into independent spatial components," Human brain mapping, vol. 6, no. 3, pp. 160–188, 1998.

[56] D. L. Donoho, "Compressed sensing," IEEE Transactions on information theory, vol. 52, no. 4, pp. 1289–1306, 2006.

[57] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 367, no. 1906, pp. 4273–4293, 2009.

[58] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing mri," IEEE signal processing magazine, vol. 25, no. 2, pp. 72–82, 2008.

[59] E. Ising, "Beitrag zur theorie des ferromagnetismus," Zeitschrift für Physik, vol. 31, no. 1, pp. 253–258, 1925.

[60] L. Onsager, "Crystal statistics. i. a two-dimensional model with an order-disorder transition," Physical Review, vol. 65, no. 3-4, p. 117, 1944.

[61] P. W. Anderson, "Absence of diffusion in certain random lattices," Physical review, vol. 109, no. 5, p. 1492, 1958.

[62] P. W. Anderson, "Localized magnetic states in metals," Physical Review, vol. 124, no. 1, p. 41, 1961.

[63] J. L. Lebowitz and O. Penrose, "Rigorous treatment of the van der waals-maxwell theory of the liquid-vapor transition," Journal of Mathematical Physics, vol. 7, no. 1, pp. 98–113, 1966.

[64] S. Kirkpatrick and D. Sherrington, "Infinite-ranged models of spin-glasses," Physical Review B, vol. 17, no. 11, p. 4384, 1978.

[65] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proceedings of the national academy of sciences, vol. 79, no. 8, pp. 2554–2558, 1982.

[66] D. J. Amit, Modeling brain function: The world of attractor neural networks. Cambridge university press, 1992.

[67] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," Cognitive science, vol. 9, no. 1, pp. 147–169, 1985.

[68] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," tech. rep., COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986.

[69] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," nature, vol. 323, no. 6088, p. 533, 1986.

[70] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural computation, vol. 14, no. 8, pp. 1771–1800, 2002.

[71] M. Welling, M. Rosen-Zvi, and G. E. Hinton, "Exponential family harmoniums with an application to information retrieval," in Advances in neural information processing systems, pp. 1481–1488, 2005.

[72] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in Proceedings of the 24th international conference on Machine learning, pp. 791–798, ACM, 2007.

[73] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[74] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," Journal of Machine Learning Research, vol. 11, no. Feb, pp. 625–660, 2010.

[75] A. Fischer and C. Igel, "An introduction to restricted boltzmann machines," in Iberoamerican Congress on Pattern Recognition, pp. 14–36, Springer, 2012.

[76] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[77] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

[78] M. Mezard and A. Montanari, Information, physics, and computation. Oxford University Press, 2009.

[79] M. Welling and Y. W. Teh, "Approximate inference in boltzmann machines," Artificial Intelligence, vol. 143, no. 1, pp. 19–50, 2003.

[80] M. Mezard and T. Mora, "Constraint satisfaction problems and neural networks: A statistical physics perspective," Journal of Physiology-Paris, vol. 103, no. 1-2, pp. 107–113, 2009.

[81] V. Sessak and R. Monasson, "Small-correlation expansions for the inverse ising problem," Journal of Physics A: Mathematical and Theoretical, vol. 42, no. 5, p. 055001, 2009.

[82] S. Cocco and R. Monasson, "Adaptive cluster expansion for inferring boltzmann machines with noisy data," Physical review letters, vol. 106, no. 9, p. 090601, 2011.

[83] S. Cocco, S. Leibler, and R. Monasson, "Neuronal couplings between retinal ganglion cells inferred by efficient inverse statistical physics methods," Proceedings of the National Academy of Sciences, vol. 106, no. 33, pp. 14058–14062, 2009.

[84] O. Marre, S. El Boustani, Y. Frégnac, and A. Destexhe, "Prediction of spatiotemporal patterns of neural activity from pairwise correlations," Physical review letters, vol. 102, no. 13, p. 138101, 2009.

[85] G. Tavoni, U. Ferrari, F. P. Battaglia, S. Cocco, and R. Monasson, "Functional coupling networks inferred from prefrontal cortex activity show experience-related effective plasticity," Network Neuroscience, vol. 1, no. 3, pp. 275–301, 2017.

[86] L. Posani, S. Cocco, K. Ježek, and R. Monasson, "Functional connectivity models for decoding of spatial representations from hippocampal ca1 recordings," Journal of computational neuroscience, vol. 43, no. 1, pp. 17–33, 2017.

[87] L. Meshulam, J. L. Gauthier, C. D. Brody, D. W. Tank, and W. Bialek, "Collective behavior of place and non-place neurons in the hippocampal network," Neuron, vol. 96, no. 5, pp. 1178–1191, 2017.

[88] L. Posani, S. Cocco, and R. Monasson, "Integration and multiplexing of positional and contextual information by the hippocampal network," bioRxiv, p. 269340, 2018.

[89] S. Cocco, R. Monasson, L. Posani, and G. Tavoni, "Functional networks from inverse modeling of neural population activity," Current Opinion in Systems Biology, vol. 3, pp. 103–110, 2017.

[90] C. Gardella, O. Marre, and T. Mora, "Modeling the correlated activity of neural populations: A review," arXiv preprint arXiv:1806.08167, 2018.

[91] W. Bialek, A. Cavagna, I. Giardina, T. Mora, E. Silvestri, M. Viale, and A. M. Walczak, "Statistical mechanics for natural flocks of birds," Proceedings of the National Academy of Sciences, 2012.

[92] T. Bury, "Market structure explained by pairwise interactions," Physica A: Statistical Mechanics and its Applications, vol. 392, no. 6, pp. 1375–1385, 2013.

[93] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814, 2010.

[94] A. Barra, A. Bernacchia, E. Santucci, and P. Contucci, "On the equivalence of hopfield networks and boltzmann machines," Neural Networks, vol. 34, pp. 1–9, 2012.

[95] N. Le Roux and Y. Bengio, "Representational power of restricted boltzmann machines and deep belief networks," Neural computation, vol. 20, no. 6, pp. 1631–1649, 2008.

[96] R. M. Neal, "Annealed importance sampling," Statistics and computing, vol. 11, no. 2, pp. 125–139, 2001.

[97] R. Salakhutdinov and I. Murray, "On the quantitative analysis of deep belief networks," in Proceedings of the 25th international conference on Machine learning, pp. 872–879, ACM, 2008.

[98] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in Proceedings of COMPSTAT'2010, pp. 177–186, Springer, 2010.

[99] W. Krauth, Statistical mechanics: algorithms and computations, vol. 13. OUP Oxford, 2006.

[100] Y. Bengio and O. Delalleau, "Justifying and generalizing contrastive divergence," Neural computation, vol. 21, no. 6, pp. 1601–1621, 2009.

[101] A. Fischer and C. Igel, "Empirical analysis of the divergence of gibbs sampling based learning algorithms for restricted boltzmann machines," in International Conference on Artificial Neural Networks, pp. 208–217, Springer, 2010.

[102] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, "Tempered markov chain monte carlo for training of restricted boltzmann machines," in Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 145–152, 2010.

[103] T. Tieleman, "Training restricted boltzmann machines using approximations to the likelihood gradient," in Proceedings of the 25th international conference on Machine learning, pp. 1064–1071, ACM, 2008.

[104] L. Younes, "On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates," Stochastics: An International Journal of Probability and Stochastic Processes, vol. 65, no. 3-4, pp. 177–228, 1999.

[105] G. Desjardins, A. Courville, and Y. Bengio, "Adaptive parallel tempering for stochastic maximum likelihood learning of rbms," arXiv preprint arXiv:1012.3476, 2010.

[106] K. Cho, T. Raiko, and A. Ilin, "Parallel tempering is efficient for learning restricted boltzmann machines," in Neural Networks (IJCNN), The 2010 International Joint Conference on, pp. 1–8, IEEE, 2010.

[107] Y. Sugita and Y. Okamoto, "Replica-exchange molecular dynamics method for protein folding," Chemical physics letters, vol. 314, no. 1-2, pp. 141–151, 1999.

[108] J. R. Anderson and C. Peterson, "A mean field theory learning algorithm for neural networks," Complex Systems, vol. 1, pp. 995–1019, 1987.

[109] S. Cocco and R. Monasson, "Adaptive cluster expansion for the inverse ising problem: convergence, algorithm and tests," Journal of Statistical Physics, vol. 147, no. 2, pp. 252–314, 2012.

[110] J. P. Barton, E. De Leonardis, A. Coucke, and S. Cocco, "Ace: adaptive cluster expansion for maximum entropy graphical model inference," Bioinformatics, vol. 32, no. 20, pp. 3089–3097, 2016.

[111] M. Welling and G. E. Hinton, "A new learning algorithm for mean field boltzmann machines," in International Conference on Artificial Neural Networks, pp. 351–357, Springer, 2002.

[112] M. Gabrié, E. W. Tramel, and F. Krzakala, "Training restricted boltzmann machine via the? thouless-anderson-palmer free energy," in Advances in Neural Information Processing Systems, pp. 640–648, 2015.

[113] E. W. Tramel, M. Gabrié, A. Manoel, F. Caltagirone, and F. Krzakala, "A deterministic and generalized framework for unsupervised learning with restricted boltzmann machines," arXiv preprint arXiv:1702.03260, 2017.

[114] J. Sohl-Dickstein, P. B. Battaglino, and M. R. DeWeese, "New method for parameter estimation in probabilistic models: minimum probability flow," Physical review letters, vol. 107, no. 22, p. 220601, 2011.

[115] M. Ekeberg, T. Hartonen, and E. Aurell, "Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences," Journal of Computational Physics, vol. 276, pp. 341–356, 2014.

[116] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[117] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "Smem algorithm for mixture models," Neural computation, vol. 12, no. 9, pp. 2109–2128, 2000.

[118] K. Cho, T. Raiko, and A. T. Ihler, "Enhanced gradient and adaptive learning rate for training restricted boltzmann machines," in Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 105–112, 2011.

[119] G. E. Hinton, "A practical guide to training restricted boltzmann machines," in Neural networks: Tricks of the trade, pp. 599–619, Springer, 2012.

[120] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," Mathematical programming, vol. 45, no. 1-3, pp. 503–528, 1989.

[121] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in International conference on machine learning, pp. 1139–1147, 2013.

[122] G. Montavon and K.-R. Müller, "Deep boltzmann machines and the centering trick," in Neural Networks: Tricks of the Trade, pp. 621–637, Springer, 2012.

[123] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," Journal of statistical planning and inference, vol. 90, no. 2, pp. 227–244, 2000.

[124] R. Salakhutdinov, "Learning deep boltzmann machines using adaptive mcmc," in Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 943–950, 2010.

[125] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," in Advances in neural information processing systems, pp. 3738–3746, 2016.

[126] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks," arXiv preprint arXiv:1701.04722, 2017.

[127] W. A. Little, "The existence of persistent states in the brain," in From High-Temperature Superconductivity to Microminiature Refrigeration, pp. 145–164, Springer, 1974.

[128] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Spin-glass models of neural networks," Physical Review A, vol. 32, no. 2, p. 1007, 1985.

[129] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Statistical mechanics of neural networks near saturation," Annals of physics, vol. 173, no. 1, pp. 30–67, 1987.

[130] M. Mézard, G. Parisi, and M. Virasoro, Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications, vol. 9. World Scientific Publishing Company, 1987.

[131] M. Mézard, "Mean-field message-passing equations in the hopfield model and its generalizations," Physical Review E, vol. 95, no. 2, p. 022117, 2017.

[132] W. Gerstner and J. L. van Hemmen, "Associative memory in a network of 'spiking'neurons," Network: Computation in Neural Systems, vol. 3, no. 2, pp. 139–164, 1992.

[133] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Information storage in neural networks with low levels of activity," Physical Review A, vol. 35, no. 5, p. 2293, 1987.

[134] M. V. Tsodyks and M. V. Feigel'man, "The enhanced storage capacity in neural networks with low activity level," EPL (Europhysics Letters), vol. 6, no. 2, p. 101, 1988.

[135] M. Tsodyks, "Associative memory in asymmetric diluted network with low level of activity," EPL (Europhysics Letters), vol. 7, no. 3, p. 203, 1988.

[136] L. Personnaz, I. Guyon, and G. Dreyfus, "Collective computational properties of neural networks: New learning mechanisms," Physical Review A, vol. 34, no. 5, p. 4217, 1986.

[137] I. Kanter and H. Sompolinsky, "Associative recall of memory without errors," Physical Review A, vol. 35, no. 1, p. 380, 1987.

[138] E. Agliari, A. Barra, A. Galluzzi, F. Guerra, and F. Moauro, "Multitasking associative networks," Physical review letters, vol. 109, no. 26, p. 268101, 2012.

[139] E. Agliari, A. Annibale, A. Barra, A. Coolen, and D. Tantari, "Immune networks: multitasking capabilities near saturation," Journal of Physics A: Mathematical and Theoretical, vol. 46, no. 41, p. 415003, 2013.

[140] P. Sollich, D. Tantari, A. Annibale, and A. Barra, "Extensive parallel processing on scale-free networks," Physical review letters, vol. 113, no. 23, p. 238106, 2014.

[141] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," arXiv preprint arXiv:1211.5590, 2012.

[142] G. Parisi, "Order parameter for spin-glasses," Physical Review Letters, vol. 50, no. 24, p. 1946, 1983.

[143] M. Mézard, G. Parisi, N. Sourlas, G. Toulouse, and M. Virasoro, "Nature of the spin-glass phase," Physical review letters, vol. 52, no. 13, p. 1156, 1984.

[144] A. Decelle, G. Fissore, and C. Furtlehner, "Spectral dynamics of learning in restricted boltzmann machines," EPL (Europhysics Letters), vol. 119, no. 6, p. 60001, 2017.

[145] A. Decelle, G. Fissore, and C. Furtlehner, "Thermodynamics of restricted boltzmann machines and related learning dynamics," arXiv preprint arXiv:1803.01960, 2018.

[146] M. J. Macias, V. Gervais, C. Civera, and H. Oschkinat, "Structural analysis of ww domains and design of a ww prototype," Nature Structural and Molecular Biology, vol. 7, no. 5, p. 375, 2000.

[147] W. P. Russ, D. M. Lowery, P. Mishra, M. B. Yaffe, and R. Ranganathan, "Natural-like function in artificial ww domains," Nature, vol. 437, no. 7058, pp. 579–583, 2005.

[148] C. Levinthal, "How to fold graciously," Mossbauer spectroscopy in biological systems, vol. 67, pp. 22–24, 1969.

[149] C. B. Anfinsen, "Principles that govern the folding of protein chains," Science, vol. 181, no. 4096, pp. 223–230, 1973.

[150] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, "How fast-folding proteins fold," Science, vol. 334, no. 6055, pp. 517–520, 2011.

[151] K. A. Dill and J. L. MacCallum, "The protein-folding problem, 50 years on," science, vol. 338, no. 6110, pp. 1042–1046, 2012.

[152] S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. E. Dawid, and A. Kolinski, "Coarse-grained protein models and their applications," Chemical Reviews, vol. 116, no. 14, pp. 7898–7936, 2016.

[153] C. Dominguez, R. Boelens, and A. M. Bonvin, "Haddock: a protein- protein docking approach based on biochemical or biophysical information," Journal of the American Chemical Society, vol. 125, no. 7, pp. 1731–1737, 2003.

[154] D. Schneidman-Duhovny, Y. Inbar, R. Nussinov, and H. J. Wolfson, "Patchdock and symmdock: servers for rigid and symmetric docking," Nucleic acids research, vol. 33, no. suppl_2, pp. W363–W367, 2005.

[155] D. D. Boehr, R. Nussinov, and P. E. Wright, "The role of dynamic conformational ensembles in biomolecular recognition," Nature chemical biology, vol. 5, no. 11, p. 789, 2009.

[156] A. Goate, M.-C. Chartier-Harlin, M. Mullan, J. Brown, F. Crawford, L. Fidani, L. Giuffra, A. Haynes, N. Irving, L. James, et al., "Segregation of a missense mutation in the amyloid precursor protein gene with familial alzheimer's disease," Nature, vol. 349, no. 6311, p. 704, 1991.

[157] J. Wang, B. J. Gu, C. L. Masters, and Y.-J. Wang, "A systemic view of alzheimer disease—insights from amyloid-β metabolism beyond the brain," Nature Reviews Neurology, vol. 13, no. 10, p. 612, 2017.

[158] M. R. Arkin, Y. Tang, and J. A. Wells, "Small-molecule inhibitors of protein-protein interactions: progressing toward the reality," Chemistry & biology, vol. 21, no. 9, pp. 1102–1114, 2014.

[159] D. E. Scott, A. R. Bayly, C. Abell, and J. Skidmore, "Small molecules, big targets: drug discovery faces the protein–protein interaction challenge," Nature Reviews Drug Discovery, vol. 15, no. 8, p. 533, 2016.

[160] M. Bakail and F. Ochsenbein, "Targeting protein–protein interactions, a wide open field for drug design," Comptes Rendus Chimie, vol. 19, no. 1-2, pp. 19–27, 2016.

[161] G. P. Smith and V. A. Petrenko, "Phage display," Chemical reviews, vol. 97, no. 2, pp. 391–410, 1997.

[162] B. I. Dahiyat and S. L. Mayo, "De novo protein design: fully automated sequence selection," Science, vol. 278, no. 5335, pp. 82–87, 1997.

[163] B. Kuhlman, G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard, and D. Baker, "Design of a novel globular protein fold with atomic-level accuracy," science, vol. 302, no. 5649, pp. 1364–1368, 2003.

[164] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker, "Protein structure prediction using rosetta," in Methods in enzymology, vol. 383, pp. 66–93, Elsevier, 2004.

[165] O. Khersonsky and S. J. Fleishman, "Why reinvent the wheel? building new proteins based on ready-made parts," Protein Science, vol. 25, no. 7, pp. 1179–1187, 2016.

[166] P.-S. Huang, S. E. Boyken, and D. Baker, "The coming of age of de novo protein design," Nature, vol. 537, no. 7620, p. 320, 2016.

[167] S. Gonen, F. DiMaio, T. Gonen, and D. Baker, "Design of ordered two-dimensional arrays mediated by noncovalent protein-protein interfaces," Science, vol. 348, no. 6241, pp. 1365–1368, 2015.

[168] J. B. Bale, S. Gonen, Y. Liu, W. Sheffler, D. Ellis, C. Thomas, D. Cascio, T. O. Yeates, T. Gonen, N. P. King, et al., "Accurate design of megadalton-scale two-component icosahedral protein complexes," Science, vol. 353, no. 6297, pp. 389–394, 2016.

[169] J. Nakai, M. Ohkura, and K. Imoto, "A high signal-to-noise ca 2+ probe composed of a single green fluorescent protein," Nature biotechnology, vol. 19, no. 2, p. 137, 2001.

[170] U. Consortium et al., "Uniprot: the universal protein knowledgebase," Nucleic acids research, vol. 46, no. 5, p. 2699, 2018.

[171] S. R. Eddy, "Accelerated profile hmm searches," PLoS computational biology, vol. 7, no. 10, p. e1002195, 2011.

[172] U. Göbel, C. Sander, R. Schneider, and A. Valencia, "Correlated mutations and residue contacts in proteins," Proteins: Structure, Function, and Bioinformatics, vol. 18, no. 4, pp. 309–317, 1994.

[173] J. Cheng and P. Baldi, "Improved residue contact prediction using support vector machines and a large feature set," BMC bioinformatics, vol. 8, no. 1, p. 113, 2007.

[174] M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, and T. Hwa, "Identification of direct residue contacts in protein–protein interaction by message passing," Proceedings of the National Academy of Sciences, vol. 106, no. 1, pp. 67–72, 2009.

[175] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, "Direct-coupling analysis of residue coevolution captures native contacts across many protein families," Proceedings of the National Academy of Sciences, vol. 108, no. 49, pp. E1293–E1301, 2011.

[176] S. Cocco, C. Feinauer, M. Figliuzzi, R. Monasson, and M. Weigt, "Inverse statistical physics of protein sequences: A key issues review," Reports on Progress in Physics, vol. 81, no. 3, p. 032601, 2018.

[177] H. Kamisetty, S. Ovchinnikov, and D. Baker, "Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era," Proceedings of the National Academy of Sciences, vol. 110, no. 39, pp. 15674–15679, 2013.

[178] D. T. Jones, D. W. Buchan, D. Cozzetto, and M. Pontil, "Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments," Bioinformatics, vol. 28, no. 2, pp. 184–190, 2011.

[179] S. Ovchinnikov, D. E. Kim, R. Y.-R. Wang, Y. Liu, F. DiMaio, and D. Baker, "Improved de novo structure prediction in casp 11 by incorporating coevolution information into rosetta," Proteins: Structure, Function, and Bioinformatics, vol. 84, pp. 67–75, 2016.

[180] S. Ovchinnikov, H. Park, N. Varghese, P.-S. Huang, G. A. Pavlopoulos, D. E. Kim, H. Kamisetty, N. C. Kyrpides, and D. Baker, "Protein structure determination using metagenome sequence data," Science, vol. 355, no. 6322, pp. 294–298, 2017.

[181] M. J. Skwark, D. Raimondi, M. Michel, and A. Elofsson, "Improved contact predictions using the recognition of protein like contact patterns," PLoS computational biology, vol. 10, no. 11, p. e1003889, 2014.

[182] S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, "Accurate de novo prediction of protein contact map by ultra-deep learning model," PLoS computational biology, vol. 13, no. 1, p. e1005324, 2017.

[183] D. Malinverni, S. Marsili, A. Barducci, and P. De Los Rios, "Large-scale conformational transitions and dimerization are encoded in the amino-acid sequences of hsp70 chaperones," PLoS computational biology, vol. 11, no. 6, p. e1004262, 2015.

[184] S. Ovchinnikov, H. Kamisetty, and D. Baker, "Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information," Elife, vol. 3, p. e02030, 2014.

[185] T. A. Hopf, C. P. Schärfe, J. P. Rodrigues, A. G. Green, O. Kohlbacher, C. Sander, A. M. Bonvin, and D. S. Marks, "Sequence co-evolution gives 3d contacts and structures of protein complexes," Elife, vol. 3, p. e03430, 2014.

[186] J. Yu, M. Vavrusa, J. Andreani, J. Rey, P. Tufféry, and R. Guerois, "Interevdock: a docking server to predict the structure of protein–protein interactions using evolutionary information," Nucleic acids research, vol. 44, no. W1, pp. W542–W549, 2016.

[187] J. K. Mann, J. P. Barton, A. L. Ferguson, S. Omarjee, B. D. Walker, A. Chakraborty, and T. Ndung'u, "The fitness landscape of hiv-1 gag: Advanced modeling approaches and validation of model predictions by in vitro testing," PLoS Comput Biol, vol. 10, p. e1003776, 08 2014.

[188] M. Figliuzzi, H. Jacquier, A. Schug, O. Tenaillon, and M. Weigt, "Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase tem-1," Molecular Biology and Evolution, vol. 33, no. 1, pp. 268–280, 2016.

[189] T. A. Hopf, J. B. Ingraham, F. J. Poelwijk, C. P. Schärfe, M. Springer, C. Sander, and D. S. Marks, "Mutation effects predicted from sequence co-variation," Nature biotechnology, vol. 35, no. 2, p. 128, 2017.

[190] W. Bialek and R. Ranganathan, "Rediscovering the power of pairwise interactions," arXiv preprint arXiv:0712.4397, 2007.

[191] A. Coucke, "High dimensional inference with correlated data: statistical modeling of protein sequences beyond structural prediction," 2016.

[192] M. Jäger, Y. Zhang, J. Bieschke, H. Nguyen, M. Dendle, M. E. Bowman, J. P. Noel, M. Gruebele, and J. W. Kelly, "Structure–function–folding relationship in a ww domain," Proceedings of the National Academy of Sciences, vol. 103, no. 28, pp. 10648–10653, 2006.

[193] A. Coucke, G. Uguzzoni, F. Oteri, S. Cocco, R. Monasson, and M. Weigt, "Direct coevolutionary couplings reflect biophysical residue interactions in proteins," The Journal of chemical physics, vol. 145, no. 17, p. 174102, 2016.

[194] S. W. Lockless and R. Ranganathan, "Evolutionarily conserved pathways of energetic connectivity in protein families," Science, vol. 286, no. 5438, pp. 295–299, 1999.

[195] G. Casari, C. Sander, and A. Valencia, "A method to predict functional residues in proteins," Nature Structural and Molecular Biology, vol. 2, no. 2, p. 171, 1995.

[196] D. M. Weinreich, Y. Lan, C. S. Wylie, and R. B. Heckendorn, "Should evolutionary geneticists worry about higher-order epistasis?," Current opinion in genetics & development, vol. 23, no. 6, pp. 700–707, 2013.

[197] F. J. Poelwijk, M. Socolich, and R. Ranganathan, "Learning the pattern of epistasis linking genotype and phenotype in a protein," bioRxiv, p. 213835, 2017.

[198] N. Halabi, O. Rivoire, S. Leibler, and R. Ranganathan, "Protein sectors: evolutionary units of three-dimensional structure," Cell, vol. 138, no. 4, pp. 774–786, 2009.

[199] O. Rivoire, K. A. Reynolds, and R. Ranganathan, "Evolution-based functional decomposition of proteins," PLoS computational biology, vol. 12, no. 6, p. e1004817, 2016.

[200] R. N. McLaughlin Jr, F. J. Poelwijk, A. Raman, W. S. Gosal, and R. Ranganathan, "The spatial architecture of protein function and adaptation," Nature, vol. 491, no. 7422, p. 138, 2012.

[201] R. G. Smock, O. Rivoire, W. P. Russ, J. F. Swain, S. Leibler, R. Ranganathan, and L. M. Gierasch, "An interdomain sector mediating allostery in hsp70 molecular chaperones," Molecular systems biology, vol. 6, no. 1, p. 414, 2010.

[202] T. Teşileanu, L. J. Colwell, and S. Leibler, "Protein sectors: Statistical coupling analysis versus conservation," PLoS computational biology, vol. 11, no. 2, p. e1004091, 2015.

[203] E. Shakhnovich and A. Gutin, "Enumeration of all compact conformations of copolymers with random sequence of links," The Journal of Chemical Physics, vol. 93, no. 8, pp. 5967–5971, 1990.

[204] L. Mirny and E. Shakhnovich, "Protein folding theory: From lattice to all-atom models," Annual Review of Biophysics and Biomolecular Structure, vol. 30, no. 1, pp. 361–396, 2001. PMID: 11340064.

[205] H. Jacquin, A. Gilson, E. Shakhnovich, S. Cocco, and R. Monasson, "Benchmarking inverse statistical approaches for protein structure and design with exactly solvable models," PLoS computational biology, vol. 12, no. 5, p. e1004889, 2016.

[206] M. Sudol, H. I. Chen, C. Bougeret, A. Einbond, and P. Bork, "Characterization of a novel protein-binding module—the ww domain," FEBS letters, vol. 369, no. 1, pp. 67–71, 1995.

[207] P. Ascenzi, A. Bocedi, M. Bolognesi, A. Spallarossa, M. Coletta, R. Cristofaro, and E. Menegatti, "The bovine basic pancreatic trypsin inhibitor (kunitz inhibitor): a milestone protein," Current Protein and Peptide Science, vol. 4, no. 3, pp. 231–251, 2003.

[208] B. Bukau and A. L. Horwich, "The hsp70 and hsp60 chaperone machines," Cell, vol. 92, no. 3, pp. 351–366, 1998.

[209] S. Miyazawa and R. L. Jernigan, "Residue–residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading," Journal of molecular biology, vol. 256, no. 3, pp. 623–644, 1996.

[210] M. Sudol and T. Hunter, "New wrinkles for an old domain," Cell, vol. 103, no. 7, pp. 1001–1004, 2000.

[211] X. Espanel and M. Sudol, "A single point mutation in a group i ww domain shifts its specificity to that of group ii ww domains," Journal of Biological Chemistry, vol. 274, no. 24, pp. 17284–17289, 1999.

[212] D. M. Fowler, C. L. Araya, S. J. Fleishman, E. H. Kellogg, J. J. Stephany, D. Baker, and S. Fields, "High-resolution mapping of protein sequence-function relationships," Nature methods, vol. 7, no. 9, p. 741, 2010.

[213] Y. Kato, M. Ito, K. Kawai, K. Nagata, and M. Tanokura, "Determinants of ligand specificity in groups i and iv ww domains as studied by surface plasmon resonance and model building," Journal of Biological Chemistry, vol. 277, no. 12, pp. 10173–10177, 2002.

[214] L. Otte, U. Wiedemann, B. Schlegel, J. R. Pires, M. Beyermann, P. Schmieder, G. Krause, R. Volkmer-Engert, J. Schneider-Mergener, and H. Oschkinat, "Ww domain sequence activity relationships identified using ligand recognition propensities of 42 ww domains," Protein Science, vol. 12, no. 3, pp. 491–500, 2003.

[215] H. Shigetomi, A. Onogi, H. Kajiwara, S. Yoshida, N. Furukawa, S. Haruta, Y. Tanase, S. Kanayama, T. Noguchi, Y. Yamada, et al., "Anti-inflammatory actions of serine protease inhibitors containing the kunitz domain," Inflammation research, vol. 59, no. 9, pp. 679–687, 2010.

[216] M. S. Bajaj, J. J. Birktoft, S. A. Steer, and S. P. Bajaj, "Structure and biology of tissue factor pathway inhibitor," Thrombosis and haemostasis, vol. 86, no. 04, pp. 959–972, 2001.

[217] E. Fries and A. M. Blom, "Bikunin—not just a plasma proteinase inhibitor," The international journal of biochemistry & cell biology, vol. 32, no. 2, pp. 125–137, 2000.

[218] M. Levitt and A. Warshel, "Computer simulation of protein folding," Nature, vol. 253, no. 5494, p. 694, 1975.

[219] T. A. Hopf, L. J. Colwell, R. Sheridan, B. Rost, C. Sander, and D. S. Marks, "Three-dimensional structures of membrane proteins from genomic sequencing," Cell, vol. 149, no. 7, pp. 1607–1621, 2012.

[220] S. Cocco, R. Monasson, and M. Weigt, "From principal component to direct coupling analysis of coevolution in proteins: Low-eigenvalue modes are needed for structure prediction," PLoS computational biology, vol. 9, no. 8, p. e1003176, 2013.

[221] A. Haldane, W. F. Flynn, P. He, and R. M. Levy, "Coevolutionary landscape of kinase family proteins: Sequence probabilities and functional motifs," Biophysical journal, vol. 114, no. 1, pp. 21–31, 2018.

[222] R. D. Finn, A. Bateman, J. Clements, P. Coggill, R. Y. Eberhardt, S. R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, et al., "Pfam: the protein families database," Nucleic acids research, vol. 42, no. D1, pp. D222–D230, 2013.

[223] M. Marquart, J. Walter, J. Deisenhofer, W. Bode, and R. Huber, "The geometry of the reactive site and of the peptide groups in trypsin, trypsinogen and its complexes with inhibitors," Acta Crystallographica Section B: Structural Science, vol. 39, no. 4, pp. 480–490, 1983.

[224] M. Chen, D. R. Keene, F. K. Costa, S. H. Tahk, and D. T. Woodley, "The carboxyl terminus of type vii collagen mediates antiparallel-dimer formation and constitutes a new antigenic epitope for eba autoantibodies," Journal of Biological Chemistry, 2001.

[225] E. Kohfeldt, W. Göhring, U. Mayer, M. Zweckstetter, T. A. Holak, M.-L. Chu, and R. Timpl, "Conversion of the kunitz-type module of collagen vi into a highly active trypsin inhibitor by site-directed mutagenesis," European journal of biochemistry, vol. 238, no. 2, pp. 333–340, 1996.

[226] D. Kirchhofer, M. Peek, W. Li, J. Stamos, C. Eigenbrot, S. Kadkhodayan, J. M. Elliott, R. T. Corpuz, R. A. Lazarus, and P. Moran, "Tissue expression, protease specificity, and kunitz domain functions of hepatocyte growth factor activator inhibitor-1b (hai-1b), a new splice variant of hai-1," Journal of Biological Chemistry, vol. 278, no. 38, pp. 36341–36349, 2003.

[227] A. Grzesiak, I. Krokoszynska, D. Krowarsch, O. Buczek, M. Dadlez, and J. Otlewski, "Inhibition of six serine proteinases of the human coagulation system by mutants of bovine pancreatic trypsin inhibitor," Journal of Biological Chemistry, vol. 275, no. 43, pp. 33346–33352, 2000.

[228] H. S. Chand, A. E. Schmidt, S. P. Bajaj, and W. Kisiel, "Structure function analysis of the reactive site in the first kunitz-type domain of human tissue factor pathway inhibitor-2," Journal of Biological Chemistry, 2004.

[229] K. Merigeau, B. Arnoux, D. Perahia, K. Norris, F. Norris, and A. Ducruix, "1.2 å refinement of the kunitz-type domain from the $\alpha 3$ chain of human type vi collagen," Acta Crystallographica Section D: Biological Crystallography, vol. 54, no. 3, pp. 306–312, 1998.

[230] J. Kraut, "Serine proteases: structure and mechanism of catalysis," Annual review of biochemistry, vol. 46, no. 1, pp. 331–358, 1977.

[231] J. J. Perona and C. S. Craik, "Structural basis of substrate specificity in the serine proteases," Protein Science, vol. 4, no. 3, pp. 337–360, 1995.

[232] L. Hedstrom, "Serine protease mechanism and specificity," Chemical reviews, vol. 102, no. 12, pp. 4501–4524, 2002.

[233] J. J. Perona and C. S. Craik, "Evolutionary divergence of substrate specificity within the chymotrypsin-like serine protease fold," Journal of Biological Chemistry, vol. 272, no. 48, pp. 29987–29990, 1997.

[234] L. Hedstrom, J. J. Perona, and W. J. Rutter, "Converting trypsin to chymotrypsin: residue 172 is a substrate specificity determinant," Biochemistry, vol. 33, no. 29, pp. 8757–8763, 1994.

[235] L. Hedstrom, S. Farr-Jones, C. A. Kettner, and W. J. Rutter, "Converting trypsin to chymotrypsin: ground-state binding does not determine substrate specificity," Biochemistry, vol. 33, no. 29, pp. 8764–8769, 1994.

[236] A. J. Guseman, S. L. Speer, G. M. Perez Goncalves, and G. J. Pielak, "Surface charge modulates protein–protein interactions in physiologically relevant environments," Biochemistry, vol. 57, no. 11, pp. 1681–1684, 2018.

[237] A. Pasternak, D. Ringe, and L. Hedstrom, "Comparison of anionic and cationic trypsinogens: the anionic activation domain is more flexible in solution and differs in its mode of bpti binding in the crystal structure," Protein science, vol. 8, no. 1, pp. 253–258, 1999.

[238] J. C. Young, V. R. Agashe, K. Siegers, and F. U. Hartl, "Pathways of chaperone-mediated protein folding in the cytosol," Nature reviews Molecular cell biology, vol. 5, no. 10, p. 781, 2004.

[239] E. R. Zuiderweg, L. E. Hightower, and J. E. Gestwicki, "The remarkable multivalency of the hsp70 chaperones," Cell Stress and Chaperones, vol. 22, no. 2, pp. 173–189, 2017.

[240] P. Bork, C. Sander, and A. Valencia, "An atpase domain common to prokaryotic cell cycle proteins, sugar kinases, actin, and hsp70 heat shock proteins.," Proceedings of the National Academy of Sciences, vol. 89, no. 16, pp. 7290–7294, 1992.

[241] C. Scheufler, A. Brinker, G. Bourenkov, S. Pegoraro, L. Moroder, H. Bartunik, F. U. Hartl, and I. Moarefi, "Structure of tpr domain–peptide complexes: critical elements in the assembly of the hsp70–hsp90 multichaperone machine," Cell, vol. 101, no. 2, pp. 199–210, 2000.

[242] A. Buchberger, H. Schröder, M. Büttner, A. Valencia, and B. Bukau, "A conserved loop in the atpase domain of the dnak chaperone is essential for stable binding of grpe," Nature Structural and Molecular Biology, vol. 1, no. 2, p. 95, 1994.

[243] D. Brehmer, S. Rüdiger, C. S. Gässler, D. Klostermeier, L. Packschies, J. Reinstein, M. P. Mayer, and B. Bukau, "Tuning of chaperone activity of hsp70 proteins by modulation of nucleotide exchange," Nature Structural and Molecular Biology, vol. 8, no. 5, p. 427, 2001.

[244] K. Briknarová, S. Takayama, L. Brive, M. L. Havert, D. A. Knee, J. Velasco, S. Homma, E. Cabezas, J. Stuart, D. W. Hoyt, et al., "Structural analysis of bag1 cochaperone and its interactions with hsc70 heat shock protein," Nature Structural and Molecular Biology, vol. 8, no. 4, p. 349, 2001.

[245] H. Sondermann, C. Scheufler, C. Schneider, J. Höhfeld, F.-U. Hartl, and I. Moarefi, "Structure of a bag/hsc70 complex: convergent functional evolution of hsp70 nucleotide exchange factors," Science, vol. 291, no. 5508, pp. 1553–1557, 2001.

[246] R. Qi, E. B. Sarbeng, Q. Liu, K. Q. Le, X. Xu, H. Xu, J. Yang, J. L. Wong, C. Vorvis, W. A. Hendrickson, et al., "Allosteric opening of the polypeptide-binding site when an hsp70 binds atp," Nature Structural and Molecular Biology, vol. 20, no. 7, p. 900, 2013.

[247] E. B. Bertelsen, L. Chang, J. E. Gestwicki, and E. R. Zuiderweg, "Solution conformation of wild-type e. coli hsp70 (dnak) chaperone complexed with adp and substrate," Proceedings of the National Academy of Sciences, vol. 106, no. 21, pp. 8471–8476, 2009.

[248] C. J. Oldfield and A. K. Dunker, "Intrinsically disordered proteins and intrinsically disordered protein regions," Annual review of biochemistry, vol. 83, pp. 553–584, 2014.

[249] E. B. Sarbeng, Q. Liu, X. Tian, J. Yang, H. Li, J. L. Wong, L. Zhou, and Q. Liu, "A functional dnak dimer is essential for the efficient interaction with heat shock protein 40 kda (hsp40)," Journal of Biological Chemistry, pp. jbc–M114, 2015.

[250] N. Morgner, C. Schmidt, V. Beilsten-Edmands, I.-o. Ebong, N. A. Patel, E. M. Clerico, E. Kirschke, S. Daturpalli, S. E. Jackson, D. Agard, et al., "Hsp70 forms antiparallel dimers stabilized by post-translational modifications to position clients for transfer to hsp90," Cell reports, vol. 11, no. 5, pp. 759–769, 2015.

[251] S. Sinai, E. Kelsic, G. M. Church, and M. A. Novak, "Variational auto-encoding of protein sequences," arxiv:1712.03346, 2017.

[252] A. J. Riesselman, J. B. Ingraham, and D. S. Marks, "Deep generative models of genetic variation capture mutation effects," arxiv:1712.06527, 2017.

RÉSUMÉ

Les Machines de Boltzmann Restreintes (RBM) sont des modèles graphiques capables d'apprendre simultanément une distribution de probabilité et une représentation des données. Malgré leur architecture relativement simple, les RBM peuvent reproduire très fidèlement des données complexes telles que la base de données de chiffres écrits à la main MNIST. Il a par ailleurs été montré empiriquement qu'elles peuvent produire des représentations compositionnelles des données, i.e. qui décomposent les configurations en leurs différentes parties constitutives. Cependant, toutes les variantes de ce modèle ne sont pas aussi performantes les unes que les autres, et il n'y a pas d'explication théorique justifiant ces observations empiriques.

Dans la première partie de ma thèse, nous avons cherché à comprendre comment un modèle si simple peut produire des distributions de probabilité si complexes. Pour cela, nous avons analysé un modèle simplifié de RBM à poids aléatoires à l'aide de la méthode des répliques. Nous avons pu caractériser théoriquement un régime compositionnel pour les RBM, et montré sous quelles conditions (statistique des poids, choix de la fonction de transfert) ce régime peut ou ne peut pas émerger. Les prédictions qualitatives et quantitatives de cette analyse théorique sont en accord avec les observations réalisées sur des RBM entraînées sur des données réelles.

Nous avons ensuite appliqué les RBM à l'analyse et à la conception de séquences de protéines. De part leur grande taille, il est en effet très difficile de simuler physiquement les protéines, et donc de prédire leur structure et leur fonction. Il est cependant possible d'obtenir des informations sur la structure d'une protéine en étudiant la façon dont sa séquence varie selon les organismes. Par exemple, deux sites présentant des corrélations de mutations importantes sont souvent physiquement proches sur la structure. A l'aide de modèles graphiques tels que les Machine de Boltzmann, on peut exploiter ces signaux pour prédire la proximité spatiale des acides-aminés d'une séquence. Dans le même esprit, nous avons montré sur plusieurs familles de protéines que les RBM peuvent aller au-delà de la structure, et extraire des motifs étendus d'acides-aminés en coévolution qui reflètent les contraintes phylogénétiques, structurelles et fonctionnelles des protéines. De plus, on peut utiliser les RBM pour concevoir de nouvelles séquences avec des propriétés fonctionnelles putatives par recombinaison de ces motifs.

Enfin, nous avons développé de nouveaux algorithmes d'entraînement et des nouvelles formes paramétriques qui améliorent significativement la performance générative des RBM. Ces améliorations les rendent compétitives avec l'état de l'art des modèles génératifs tels que les réseaux génératifs adversariaux ou les auto-encodeurs variationels pour des données de taille intermédiaires.

MOTS CLÉS

Physique Statistique, Apprentissage Automatique, Analyse de séquence de protéines, Systèmes désordonnés, Modèles génératifs, Coévolution

ABSTRACT

Restricted Boltzmann Machines (RBM) are graphical models that learn jointly a probability distribution and a representation of data. Despite their simple architecture, they can learn very well complex data distributions such the handwritten digits data base MNIST. Moreover, they are empirically known to learn compositional representations of data, i.e. representations that effectively decompose configurations into their constitutive parts. However, not all variants of RBM perform equally well, and little theoretical arguments exist for these empirical observations.

In the first part of this thesis, we ask how come such a simple model can learn such complex probability distributions and representations. By analyzing an ensemble of RBM with random weights using the replica method, we have characterised a compositional regime for RBM, and shown under which conditions (statistics of weights, choice of transfer function) it can and cannot arise. Both qualitative and quantitative predictions obtained with our theoretical analysis are in agreement with observations from RBM trained on real data.

In a second part, we present an application of RBM to protein sequence analysis and design. Owe to their large size, it is very difficult to run physical simulations of proteins, and to predict their structure and function. It is however possible to infer information about a protein structure from the way its sequence varies across organisms. For instance, Boltzmann Machines can leverage correlations of mutations to predict spatial proximity of the sequence amino-acids. Here, we have shown on several synthetic and real protein families that provided a compositional regime is enforced, RBM can go beyond structure and extract extended motifs of coevolving amino-acids that reflect phylogenic, structural and functional constraints within proteins. Moreover, RBM can be used to design new protein sequences with putative functional properties by recombining these motifs at will.

Lastly, we have designed new training algorithms and model parametrizations that significantly improve RBM generative performance, to the point where it can compete with state-of-the-art generative models such as Generative Adversarial Networks or Variational Autoencoders on medium-scale data.

KEYWORDS

Statistical Physics, Machine Learning, Protein Sequence Analysis, Disordered Systems, Generative Models, Coevolution