



Disentangling Representations in Restricted Boltzmann Machines without Adversaries

Jorge Fernandez-De-Cossio-Diaz, Simona Cocco, Rémi Monasson

► To cite this version:

Jorge Fernandez-De-Cossio-Diaz, Simona Cocco, Rémi Monasson. Disentangling Representations in Restricted Boltzmann Machines without Adversaries. *Physical Review X*, 2023, 13 (2), pp.021003. 10.1103/PhysRevX.13.021003 . hal-04307038

HAL Id: hal-04307038

<https://hal.science/hal-04307038>

Submitted on 8 Feb 2024

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.



Distributed under a Creative Commons Attribution 4.0 International License

Disentangling Representations in Restricted Boltzmann Machines without Adversaries

Jorge Fernandez-de-Cossio-Diaz^{✉,*}, Simona Cocco[✉], and Rémi Monasson[✉]

Laboratory of Physics of the Ecole Normale Supérieure, CNRS UMR 8023 and PSL Research,
Sorbonne Université, Paris, France



(Received 21 July 2022; revised 16 January 2023; accepted 8 March 2023; published 5 April 2023)

A goal of unsupervised machine learning is to build representations of complex high-dimensional data, with simple relations to their properties. Such disentangled representations make it easier to interpret the significant latent factors of variation in the data, as well as to generate new data with desirable features. The methods for disentangling representations often rely on an adversarial scheme, in which representations are tuned to avoid discriminators from being able to reconstruct information about the data properties (labels). Unfortunately, adversarial training is generally difficult to implement in practice. Here we propose a simple, effective way of disentangling representations without any need to train adversarial discriminators and apply our approach to Restricted Boltzmann Machines, one of the simplest representation-based generative models. Our approach relies on the introduction of adequate constraints on the weights during training, which allows us to concentrate information about labels on a small subset of latent variables. The effectiveness of the approach is illustrated with four examples: the CelebA dataset of facial images, the two-dimensional Ising model, the MNIST dataset of handwritten digits, and the taxonomy of protein families. In addition, we show how our framework allows for analytically computing the cost, in terms of the log-likelihood of the data, associated with the disentanglement of their representations.

DOI: [10.1103/PhysRevX.13.021003](https://doi.org/10.1103/PhysRevX.13.021003)

Subject Areas: Computational Physics,
Statistical Physics

I. INTRODUCTION

Unsupervised learning involves mapping data points to adequate representations, where the statistical features relevant to the data distribution are encoded by latent variables [1]. Examples of unsupervised architectures include Restricted Boltzmann Machines [2], variational autoencoders [3], and generative adversarial networks [4], among others. However, the mapping between latent-variable activities and the relevant properties of the data is generally complex and not easily interpretable (Fig. 1), a phenomenon referred to as entanglement of representations in machine learning, or mixed sensitivity in computational neuroscience [5]. Entangled representations are hard to interpret and manipulate, e.g., for generating new data with the desired properties [1,6].

A stream of literature has recently focused on how to train unsupervised models to obtain disentangled representations, where information about certain properties is concentrated in some latent variables and excluded

from others [7–13], or absent altogether from representations [14,15]. Concentration of information, in turn, makes it possible to change the values of a few variables and generate data points with controlled properties [7]. In practice, learning of disentangled representations is often done in an adversarial framework through optimization of variational bounds to quantities hard to estimate, such as mutual information between the data features and some part of the representations. While conceptually appealing, this approach may be tricky to adopt from a numerical point of view, due to well-known difficulties in adversarial-based learning [16]. In addition, its complexity has prevented theoretical analysis so far, leaving important questions, such as the cost of disentangling representations, unanswered.

As a concrete illustration, which we consider later on in this work, imagine training an unsupervised model from a set of face images. Once learning is complete, the model can be used to generate many new faces, generalizing from the features in the training data. Generated images will show smiling faces, wearing eyeglasses, with bald heads; i.e., they will be characterized by a collection of attributes. From a practical point of view, disentangling the representations of those data would make it possible, in the generation process, to control and modify one of these attributes, such as smiling vs not smiling, while leaving the remaining ones (the overall shape of the face) unchanged. From a conceptual point of view, the coordinates of the

*j.cossio.diaz@gmail.com

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

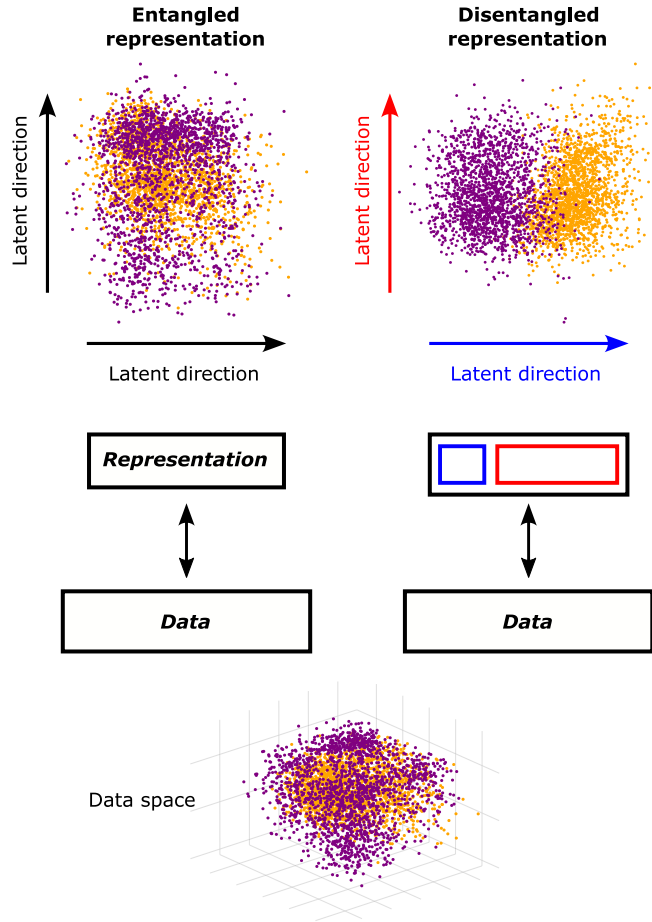


FIG. 1. Entangled vs disentangled representations. A set of high-dimensional data points (bottom) is mapped through unsupervised learning onto a latent representation (top). Data are colored in purple and orange according to a binary-valued attribute, e.g., being an odd or even number for MNIST images of handwritten digits. Left: When representations are entangled, the separation of data classes is not aligned with a single latent direction. Right: When representations are disentangled, one or few directions in latent space (blue) separate the labeled classes, while other directions are not correlated with the label (red).

representation space are explicitly related to the different attributes. Moving from one face with eyeglasses to the “same” face without corresponds to a translation of the representation vector of the face in the low-dimensional space defined by the few coordinates associated with the eyeglasses attribute, a property bearing some analogy with WORD2VEC encodings [17].

The purpose of the present work is to propose a method for disentanglement of representations, which is both effective on real data and amenable to mathematical analysis. We consider Restricted Boltzmann Machines (RBMs), a simple unsupervised generative model implementing a data–representation duality [18]. RBMs are used as building bricks of deeper networks [2] and are competitive with more complex models in various relevant situations [19–21]. We derive conditions on the RBM

parameters, which deprive all or part of the representation from information about data labels. This procedure allows us to concentrate the information about labels into a subset of latent units. Manipulation of these units then allows us to generate high-quality data with prescribed label values. Furthermore, the simplicity of our framework allows us to estimate the loss in log-likelihood resulting from the disentanglement requirement, with a deep connection with Poincaré separation theorem [22]. Informally speaking, this loss is the cost to be paid for enhanced interpretability of the machine.

Our paper is organized as follows. We first show that standard learning with RBM generically produces entangled representations on four applications chosen for their diversity and interest: (1) the CelebA dataset of face images [23] annotated with several binary attributes, (2) the two-dimensional Ising model, where configurations are annotated by the sign of their magnetizations, (3) the MNIST dataset of handwritten digits [24], where the digits represented in each image are the labels, and (4) protein-sequence families from the Pfam database [25] annotated based on their taxonomic origins. We then present how our approach learns disentangled representations and demonstrate its effectiveness when applied to the three data distributions listed above. Special emphasis is placed on the physical meaning of the unsupervised models corresponding to the Ising model case. We then calculate the costs associated with representation disentanglement.

II. REPRESENTATIONS OF COMPLEX DATA WITH RESTRICTED BOLTZMANN MACHINES ARE GENERALLY ENTANGLED

A. Unsupervised learning with RBMs

RBMs are bipartite graphical models over N visible variables $\mathbf{v} = \{v_1, v_2, \dots, v_N\}$ and M hidden (or latent) variables $\mathbf{h} = \{h_1, h_2, \dots, h_M\}$; see Fig. 2(a). Both visible and hidden variables are assumed to be Bernoulli, i.e., to take 0 or 1 values. The two layers are connected through the interaction weights $w_{i\mu}$. A RBM defines a joint probability distribution over \mathbf{v} and \mathbf{h} through

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})}, \quad (1)$$

where Z is a normalizing factor, and the energy E is given by

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^N g_i v_i - \sum_{\mu=1}^M \theta_{\mu} h_{\mu} - \sum_{\mu=1}^M I_{\mu}(\mathbf{v}) h_{\mu}. \quad (2)$$

The parameters g_i and θ_{μ} are local fields biasing the distributions of single units, and

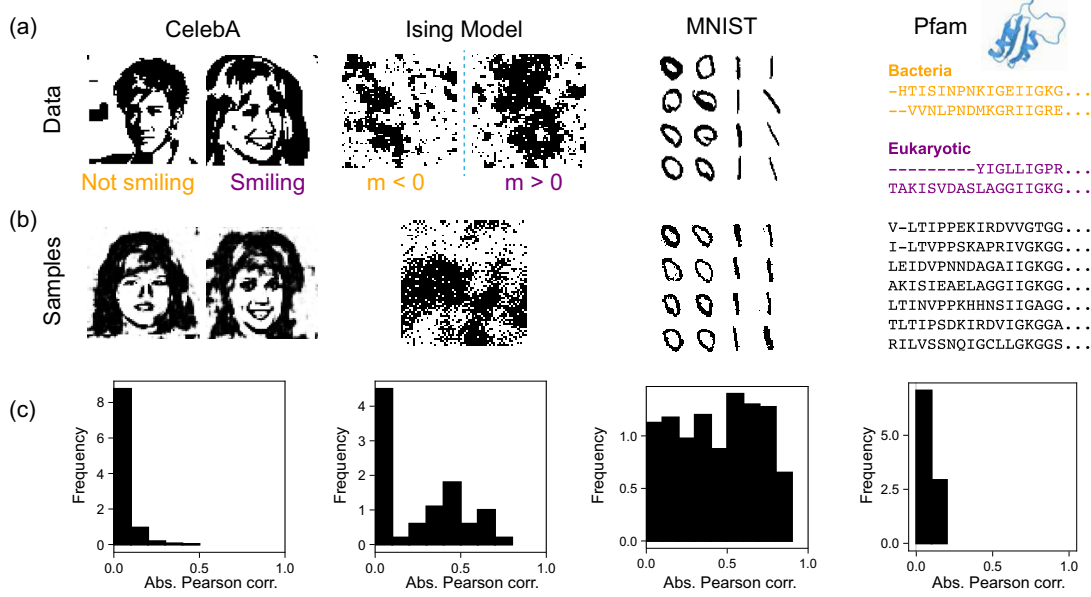


FIG. 2. Datasets considered in the paper and entanglement of representations. (a) CelebA dataset of face images [23]; two-dimensional Ising model; MNIST0/1 database of handwritten digits [24]; multiple sequence alignments from the Pfam PF00013 family of the KH domain. (b) Samples generated by different RBMs trained on each dataset. See Supplemental Material [26] Appendix A 6 for the architectures of the RBMs used in each case. (c) Histogram of the absolute value of the Pearson correlations between hidden-unit inputs and the chosen label; see Eq. (5). Smiling or not smiling for CelebA, sign of the magnetization for the Ising model, whether the digit is a 0 or 1 for MNIST0/1, and whether the KH sequence is from bacterial or eukaryotic origin.

$$I_{\mu}(\mathbf{v}) = \sum_{i=1}^N w_{i\mu} v_i \quad (3)$$

is the input received by hidden unit μ given the visible configuration.

Marginalizing over the states of the hidden units results in the likelihood $P(\mathbf{v}) = (1/Z) \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}$ of visible configurations that can be fit to the data. Given a set of data points \mathcal{D} , the weights and potential-defining parameters of the RBM are learned through gradient ascent of the dataset log-likelihood,

$$\mathcal{L} = \langle \log P(\mathbf{v}) \rangle_{\mathcal{D}}, \quad (4)$$

where the average $\langle \cdot \rangle_{\mathcal{D}}$ is taken over the data points. In practice, computing the gradient of \mathcal{L} requires us to estimate the moments of visible and/or hidden variables with respect to the model distribution [18]. Regularization of the weights can also be easily included in this approach. Details about the computation of the gradient and the training procedure implemented in this work can be found in Supplemental Material [26] Appendix A.

B. Datasets

We train the RBM on four datasets illustrated by the four columns in Fig. 2.

1. CelebA face images dataset

The CelebA dataset consists of a collection of 202 599 color images of celebrity faces, each annotated with 40 binary attributes, including whether the person is smiling, wearing glasses, has a beard, and others [23]. The images in this dataset cover large pose variations and background clutter. Figure 2(a) shows a pair of black-and-white versions of CelebA examples; see Supplemental Material [26] Fig. S1 for more examples and Supplemental Material Appendix B for processing details.

2. Two-dimensional Ising model

We next consider the Ising model [27] on a two-dimensional regular $L \times L$ square grid ($L = 32$ or 64) with uniform positive interactions between nearest-neighbor spins. The values of the interaction, or, equivalently, of the inverse temperature, are varied to explore both paramagnetic (weak interactions) and ferromagnetic (strong interactions) regimes. The data are configurations of the Ising model generated by Monte Carlo sampling and labeled according to the sign u of its magnetization m , i.e., the differences between the numbers of + [black dots in Fig. 2(a)] and − spins (white dots).

3. MNIST handwritten digits

The MNIST dataset [24] consists of a collection of 70 000 images of 28×28 pixels each, labeled by the identity of the 0–9 handwritten digit they represent. We

show 16 of them in Fig. 2(a). We hereafter consider in particular (1) MNIST0/1, a simplified version of MNIST consisting only of images of the digits 0 and 1, with binary labels $u = 0, 1$, and (2) MNIST0/1/2/3, the set of all images of digits from 0 to 3, with four-state labels u . In Supplemental Material [26] Fig. S6, we also consider an additional example consisting of zero digits only in black or white backgrounds (see Sec. VIB 3).

4. Pfam database of protein family sequences

Last of all, we consider protein families in the Pfam sequence database [25]. A protein family consists of a collection of homologous protein sequences from different organisms, i.e., sharing common evolutionary origins and common functional or structural features. As an illustration, Fig. 2(a) sketches some sequences of the K -homology (KH) domain found in nucleic-acid binding proteins. Many families include sequences issued from prokaryotic and eukaryotic organisms, and we use this classification as the label u for sequences in the dataset.

C. RBMs generically learn entangled representations

We train RBMs with 200–400 binary hidden units on CelebA images, two-dimensional Ising model configurations, MNIST0/1 digits, and KH domain protein sequences (see Supplemental Material [26] Appendix A 6 for details). Consistent with previous results on similar datasets [19,20,28,29], the RBMs accurately fit the data and generate high-quality samples in the four cases; see Fig. 2(b). In addition, training simple classifiers to predict the label from the hidden inputs of the models gives areas under the curve (AUC) > 0.9 for all cases; see Supplemental Material [26] Appendix E for details and Fig. S4. These results demonstrate that the RBM automatically captures information relevant to the labels of interest. We emphasize that in all cases the RBM does not have access to the labels during training.

We plot in Fig. 2(c) the histogram of Pearson correlations between the label and hidden-unit inputs,

$$\rho_\mu = \frac{\langle u(\mathbf{v}) I_\mu(\mathbf{v}) \rangle_{\mathcal{D}} - \langle u(\mathbf{v}) \rangle_{\mathcal{D}} \langle I_\mu(\mathbf{v}) \rangle_{\mathcal{D}}}{\sqrt{\langle I_\mu(\mathbf{v})^2 \rangle_{\mathcal{D}} - \langle I_\mu(\mathbf{v}) \rangle_{\mathcal{D}}^2} \sqrt{\langle u(\mathbf{v})^2 \rangle_{\mathcal{D}} - \langle u(\mathbf{v}) \rangle_{\mathcal{D}}^2}}. \quad (5)$$

For some datasets (e.g., KH sequences), hidden units have low correlations to the label. Changing the label identity of the generated data requires us to act on the states of all these hidden units in a concerted manner. In other cases, such as the Ising model and MNIST, a number of units exhibit higher correlations with the labels; see right tails of distributions in Fig. 2(c). However, as the label information captured by the RBM is distributed among these units, manipulating the few most correlated units is not sufficient to define the label of generated data; see Supplemental Material [26] Fig. S2.

Although a precise definition of disentangled representation learning may be debated [6,13], it is generally agreed that interesting features should map to single, or few dimensions, in latent space; see Fig. 1 [1]. As we show above, standard training of a RBM fails to produce disentangled representations.

III. LEARNING OF DISENTANGLED REPRESENTATIONS

Our strategy for disentangling and manipulating representations is to drastically alter the distribution of correlations between hidden units and labels [Fig. 2(c)] by imposing appropriate constraints on the interaction weights throughout the learning process.

Ideally, constraints should impose that mutual information, rather than correlations, vanishes. Because of the difficulty in computing mutual information, we focus on correlations at different orders in the hidden inputs, as they offer a good compromise between computational efficiency and performance. Focusing on inputs I_μ rather than on latent variables h_μ follows a twofold motivation. First, the constraints on the weights $w_{i\mu}$ resulting from the vanishing requirements on the correlations are simpler to interpret and fulfill from a computational point of view. Second, given a data configuration \mathbf{v} , h_μ is a stochastic variable conditioned to I_μ . By virtue of the data processing inequality [30], the mutual information between labels u and inputs I_μ upper bounds its counterpart between u and h_μ , and enforcing low mutual information between labels and inputs therefore immediately implies that latent variables are not informative about labels.

Two objectives can be pursued.

- (A) Approximating as best as possible the data distribution, while removing as much information as possible about their labels. This can be achieved by an architecture in which all hidden units are under strong constraints; see Fig. 3(a). Objective A leads to a generic model distribution in which label-associated features are blurred; i.e., it is hard to tell whether they are present or absent. Conversely, the other “orthogonal” features are well captured by this RBM model.

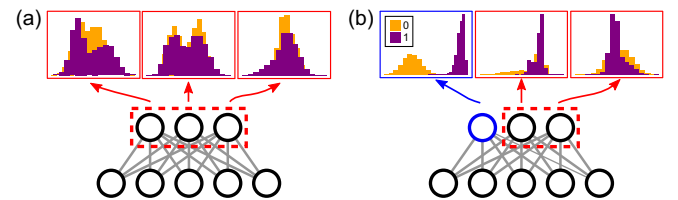


FIG. 3. Model schema. (a) Constraints imposed on all hidden units promote overlapping hidden input distributions of the two classes. (b) Constraints imposed on a subset of hidden units (red) promote class separation on the remaining hidden units (blue).

- (B) Reproducing as best as possible the data distribution, while concentrating as much information as possible about their labels on one (or few) hidden units. This can be achieved by an architecture in which a few hidden units are left unconstrained and are referred to as released, while all the other ones are under strong constraints; see Fig. 3(b). Objective B defines a model distribution, in which label-associated features are either present or absent, as in the training data. In addition, the representations can be easily manipulated to bias data generation, e.g., to morph one configuration into another one in which the label value has changed but other features have not.

For the sake of simplicity, we present the approach in the case of binary labels $u = 0, 1$ (equivalently, $u = \pm 1$). An extension to labels with more than two values is immediate and is discussed in the applications.

A. Fully constrained RBMs

Following objective A, we demand that all hidden-unit inputs I_μ are uncorrelated with the labels u across the data. The corresponding architecture is sketched in Fig. 3(a). A RBM trained under these constraints defines a distribution, in which information about the label has been degraded, if not fully erased, but the other data-defining features are affected as little as possible.

1. Linear constraints

In its simplest formulation, the approach considers only linear correlations in the inputs. The constraint $\rho_\mu = 0$ [see Eq. (5)] can be rewritten as

$$\sum_{i=1}^N q_i^{(1)} w_{i\mu} = 0, \quad (6)$$

with

$$q_i^{(1)} = \langle u(\mathbf{v}) v_i \rangle_{\mathcal{D}} - \langle u(\mathbf{v}) \rangle_{\mathcal{D}} \langle v_i \rangle_{\mathcal{D}}. \quad (7)$$

The N -dimensional vector $\mathbf{q}^{(1)}$ is parallel to the line joining the centers of mass of the clouds of data points associated with, respectively, $u = 0$ and $u = 1$; see Fig. 4(a). Imposing $\rho_\mu = 0$ for all $\mu = 1, \dots, M$ is thus equivalent to looking for the RBM maximizing the log-likelihood \mathcal{L} in Eq. (4) under the constraints that all M weight vectors \mathbf{w}_μ are orthogonal to $\mathbf{q}^{(1)}$; this can be easily done by projecting the gradient of \mathcal{L} onto the space orthogonal to $\mathbf{q}^{(1)}$ after each update of the weights (see Supplemental Material [26] Appendix A for details). In other words, the RBM is blind to the direction $\mathbf{q}^{(1)}$ separating the clouds and is modeling only the statistical features of the data in the $(N-1)$ -dimensional space orthogonal to $\mathbf{q}^{(1)}$.

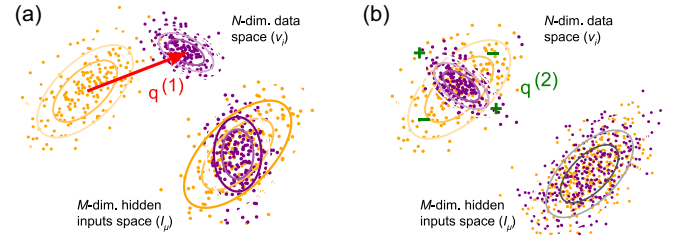


FIG. 4. First- and second-order constraints. (a) The first-order constraint (6) ensures that the classes have the same means in input space by imposing orthogonality of the weights to the vector separating their centers of mass in data space (red). (b) Second-order constraints (9) ensure that the two classes have the same covariance in input space.

The consequences of $\mathbf{w}_\mu \perp \mathbf{q}^{(1)}$ can be phrased in an adversarial context. Imagine a linear discriminator is trying to predict the labels $u(\mathbf{v})$ of data configurations \mathbf{v} based on the M -dimensional sets of inputs $I_\mu(\mathbf{v})$. In practice, a linear discriminator is parametrized by M weights a_μ and assigns a probability $\pi(\sum_\mu a_\mu I_\mu(\mathbf{v}))$ to, say, label $u = 1$ (and probability $1 - \pi$ to $u = 0$) given \mathbf{v} , where π is some sigmoid function comprised between 0 and 1. The parameters a_μ are fitted to maximize the probability that the discriminator makes the correct prediction. In geometrical terms, this is equivalent to finding the hyperplane (orthogonal to \mathbf{a} in M dimensions) separating the classes of data points \mathbf{I} associated with $u = 0$ and $u = 1$ with the largest margin [31]. We show in the Supplemental Material [26] Appendix C that, under the conditions expressed in Eq. (6), the best linear discriminator cannot do better than random guessing of the labels. In other words, imposing constraints (6) is equivalent to demanding that no adversarial linear discriminator looking at hidden-unit inputs is able to predict the labels associated with configurations.

2. Quadratic constraints

Even if no linear discriminator can recover the label from the inputs I_μ , more complex machines, such as deep neural networks, could still be able to predict the label [32] if the mutual information between u and $\mathbf{I} = (I_1, I_2, \dots, I_M)$ is nonzero. Imposing $\rho_\mu = 0$ can be seen as a first-order approximation to the stronger condition that the mutual information (MI) between the label and the inputs vanishes, $\text{MI}(u, \mathbf{I}) = 0$. The latter implies that not only the linear correlations but also all higher-order connected moments between u and \mathbf{I} vanish. In particular, the second-order correlations

$$C_{\mu,\nu} = \langle u(\mathbf{v}) I_\mu(\mathbf{v}) I_\nu(\mathbf{v}) \rangle_{\mathcal{D}} - \langle u(\mathbf{v}) \rangle_{\mathcal{D}} \langle I_\mu(\mathbf{v}) I_\nu(\mathbf{v}) \rangle_{\mathcal{D}} \quad (8)$$

should also vanish. Setting $C_{\mu,\nu} = 0$ for all pairs μ, ν in Eq. (8) forces the two classes of data attached to $u = 0$ and $u = 1$ to have identical covariance matrices in the input space. These constraints imply that no kernel-based adversarial

discriminator, where the kernel is a quadratic function of the inputs, would be able to predict the label values (see Supplemental Material [26] Appendix C for a proof). More generally, higher-order constraints would rule out the possibility for discriminator adversaries with polynomial kernels of higher degrees to successfully classify the data [33] (see Supplemental Material [26] Appendix C)).

In practice, setting $C_{\mu,\nu} = 0$ amounts to imposing a quadratic constraint over the weight vectors:

$$\sum_{i,j=1}^N q_{i,j}^{(2)} w_{i\mu} w_{j\nu} = 0, \quad (9)$$

where the mean difference between the covariance matrices of the two classes of data is defined through

$$q_{i,j}^{(2)} = \langle u(\mathbf{v}) v_i v_j \rangle_{\mathcal{D}} - \langle u(\mathbf{v}) \rangle_{\mathcal{D}} \langle v_i v_j \rangle_{\mathcal{D}}; \quad (10)$$

see illustration in Fig. 4(b). To draw a physical analogy, the $\mathbf{q}^{(2)}$ matrix looks like the quadrupole tensor separating positive and negative charge distributions in electrostatics, while $\mathbf{q}^{(1)}$ is analogous to a dipole moment.

To implement constraints (9) in practice, we square the left-hand side of Eq. (9) and add it to the optimization objective during learning times a large penalty term; see Supplemental Material [26] Appendix A for details.

The matrix $\mathbf{q}^{(2)}$ defined in Eq. (10) is estimated on empirical data and is subject to sampling noise. In practice, from finite datasets one can extract reliable estimates only of the top components of $\mathbf{q}^{(2)}$, while the empirically observed lower components will be dominated by noise. The Marchenko-Pastur (MP) law [34] describing the spectrum of correlation matrices in the null model case of independent variables can be used to estimate the thresholds between eigenvalues dominated by noise and eigenvalues reflecting the presence of structure in the data. The MP spectrum predicts that all eigenvalues λ located in the range $[\lambda_-; \lambda_+]$ have to be discarded, with $\lambda_{\pm} = (1 \pm \sqrt{r})^2$, where r is the ratio of the numbers of variables and samples. As an example, for the MNIST0/1 dataset, we estimate $\lambda_+ \simeq 1.6$ for both 0 and 1 digits. Out of the 784 eigenvalues of $\mathbf{q}^{(2)}$, only 60 (61) are larger than this bound for the 0s dataset (1s). The above discussion suggests replacing the full matrix $\mathbf{q}^{(2)}$ with a low-rank approximation focusing on the top components only. A lower-rank version of $\mathbf{q}^{(2)}$ also implies that the weights have more degrees of freedom, since Eq. (9) does not affect the weights components belonging to the kernel of $\mathbf{q}^{(2)}$. In practice, penalizing the squared norm of the left-hand side of Eq. (9) during training automatically places more weight on constraints associated with the top components of $\mathbf{q}^{(2)}$ and neglects lower components.

B. Partially constrained RBMs

We now consider objective B. Our objective is to concentrate the information about the labels on one of a few released hidden units. For this purpose, we consider the architecture of Fig. 3(b). The weights attached to these released hidden units are unconstrained during training, while the other weights are subject to the linear or quadratic constraints in Eqs. (6) and (9), as in objective A. Informally speaking, this strategy will turn the large number of weak input-label correlations found in standard RBM representations [Fig. 2(c)] into a small number of large correlations ($\propto M$) present on the released hidden units only.

1. Manipulation of label-determining hidden units

As a consequence, the values of the released hidden units strongly affect the conditional distribution of visible configurations and act as knobs that can be manipulated to generate data with desired labels. Manipulation is carried out as follows: To lighten notations we assume that a single hidden unit, say, $\mu = 1$, is released. The value of this unit h_1 is fixed (to 0 or 1). We then sample the remaining hidden units (attached to the constrained weights) and the visible units using alternate Gibbs sampling (see Supplemental Material [26] Appendix A). The visible configurations \mathbf{v} are then distributed according to a conditional probability $P(\mathbf{v}|h_1)$ and span a class of the data corresponding to a specific label value u . Flipping h_1 to $1 - h_1$ allows us to change class and quickly morph a data configuration into the closest configuration with a flipped label.

2. Cost of disentanglement

Constraining all weight vectors (objective A) is damaging the capability of RBMs to reproduce the data distribution. The loss in performance is measured by the change in log-likelihoods of test data due to the partial erasure of information about the labels,

$$\Delta \mathcal{L}_{\text{part erasure}} = \mathcal{L}_{\text{unconstr}} - \mathcal{L}_{\text{constr}}. \quad (11)$$

In the equation above, $\mathcal{L}_{\text{constr}}$ denotes the log-likelihood of data estimated with the fully constrained RBM, and $\mathcal{L}_{\text{unconstr}}$ corresponds to the standard (unconstrained) RBM. As we see in subsequent applications, this difference is generally large.

Once one or few hidden units are released (objective B), the test log-likelihood increases to \mathcal{L}_{rel} . We define the cost for disentangling representations through

$$\Delta \mathcal{L}_{\text{disent}} = \mathcal{L}_{\text{unconstr}} - \mathcal{L}_{\text{rel}}. \quad (12)$$

This cost is guaranteed to be non-negative if both RBMs are trained with equal hyperparameters, e.g., if they have the same number of hidden units and weight regularizations.

IV. APPLICATION TO FACE IMAGES

A. Learning with standard RBMs

We first illustrate our approach on the CelebA dataset of celebrity face images [23]. Since we choose to work with binary RBMs for simplicity, we first convert the images to binary black-and-white pixels of resolution 64×64 , following a procedure similar to Ref. [35] and detailed in the Supplemental Material [26] Appendix B. Using the annotations available in the dataset, we choose the presence or absence of eyeglasses and smiling or not smiling as our labels. We compute the vector \mathbf{q} defined by Eq. (7) for each

one of these two labels. Figure 5(a) shows sample images arranged in increasing value of their projection along this vector, as well as the histograms of these projections over the dataset for each label.

Next, we train a standard RBM on this dataset. Following Ref. [35] we use 5000 hidden units (Supplemental Material [26] Appendix B). After training, we generate 10 000 samples starting from random binary configurations and running Gibbs sampling for 5000 iterations. Some sampled configurations are shown in Fig. 5(b), as well as the histogram of projections along direction \mathbf{q} . Samples are diverse and span the different classes present in the dataset,

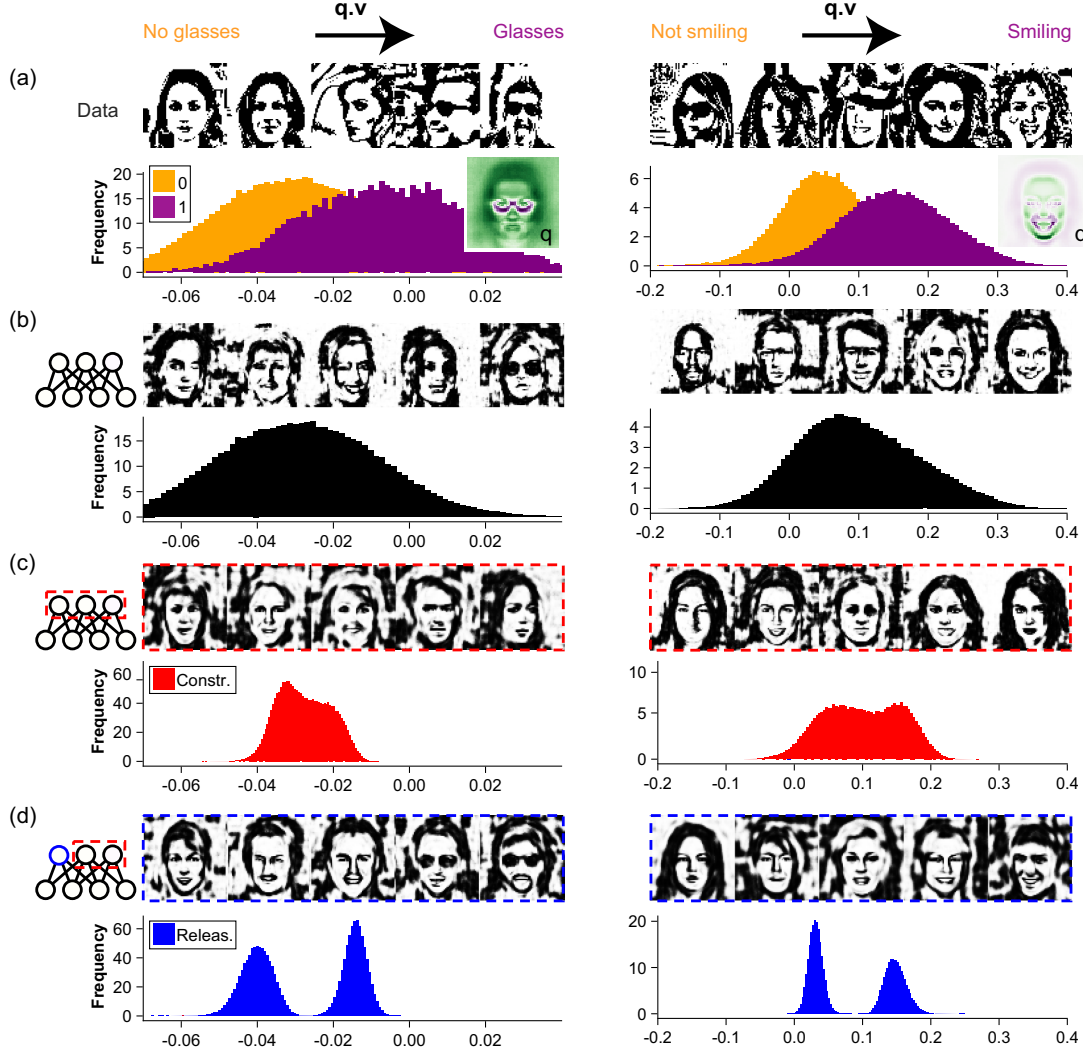


FIG. 5. Application to the CelebA dataset. Left: label corresponds to the presence or absence of eyeglasses. Right: label corresponds to smiling or not smiling. (a) Selected images from the data arranged by the value of their projection along the vector \mathbf{q} defined in Eq. (7). Below, the histogram of these projections is computed for all images in the data. The inset shows a heat map of the vector \mathbf{q} . (b) Samples generated by an unconstrained RBM and histogram of their projections on vector \mathbf{q} . (c) Samples generated by a RBM, all the hidden units of which are subject to the constraint in Eq. (6) (dashed red). The histogram (red) of projections on \mathbf{q} concentrates on intermediate values. (d) Samples generated by a RBM trained under constraint (6) acting on all but one released hidden unit (dashed blue) and histogram of projections along \mathbf{q} (blue). Details about the RBMs' architecture and training can be found in the Supplemental Material [26] Appendix A 6.

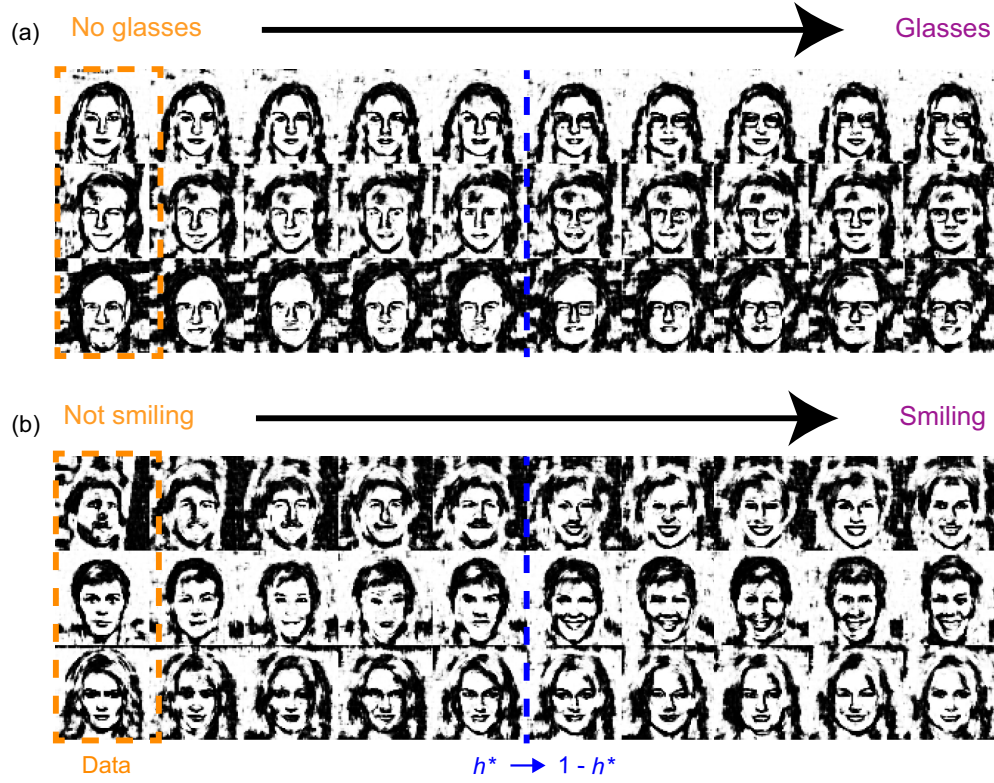


FIG. 6. Transitions between labeled classes in the CelebA dataset. RBMs are trained subject to the linear constraint acting on all but the first hidden unit denoted h^* . Samples are generated conditioned on a frozen value of h^* , which is flipped in the center of the Markov chain (indicated by the dashed blue lines). (a) Label corresponds to the “eyeglasses” attribute of CelebA. Samples are collected every three Gibbs iterations. (b) Label corresponds to the “smiling” attribute of CelebA. Samples are collected every five Gibbs iterations.

i.e., smiling or not smiling, wearing or not wearing eyeglasses, indicating that RBM is an adequate generative model for this dataset.

B. Partial erasure of information with a fully constrained RBM

We next consider a RBM with the same architecture and with constraint (6) acting on all hidden units. Figure 5(c) shows samples from such a RBM (dashed red). These samples are recognizable faces similar to the data; therefore, the model is generative. In the projection on \mathbf{q} , they concentrate on intermediate values and seem to be ambiguous with respect to the label-associated feature: Eyes seem closed or darkened in the eyeglasses case, and the mouth seems slightly open, but not entirely smiling in the second case. These findings nicely illustrate the effects of objective A.

C. Manipulating representations and face attributes with a partially constrained RBM

We now train a RBM with constraint (6) acting on all but one hidden unit, say, h^* . The weights attached to this unit are correlated with the vector $\mathbf{q}^{(1)}$ shown in Fig. 5(a) (inset). The model is generative; representative samples are shown

in Fig. 5(c), bottom panel. The projection of these samples along the \mathbf{q} direction is bimodal, with two peaks corresponding to the two values of the released hidden unit h^* . Inspecting the samples shows that h^* correlates with the attribute, as shown below, in full agreement with objective B.

The value of h^* can be manipulated during sampling to drive the Markov chain toward one class or another. We illustrate this in Fig. 6, where an initial sample from the data is sampled through this model, and the value of h^* is flipped at the midpoint of the sampling chain. As a result, the face images transition toward the expected label value. The transition is smooth: Right after the flip of h^* , most facial features are still preserved, while the one associated with the label is modified (morphing effect).

V. APPLICATION TO THE TWO-DIMENSIONAL ISING MODEL

The two-dimensional Ising model is defined by the following energy function over $N = L^2$ spin configurations $\mathbf{v} = (v_1, v_2, \dots, v_N)$,

$$E(\mathbf{v}) = - \sum_{(i,j)} v_i v_j, \quad (13)$$

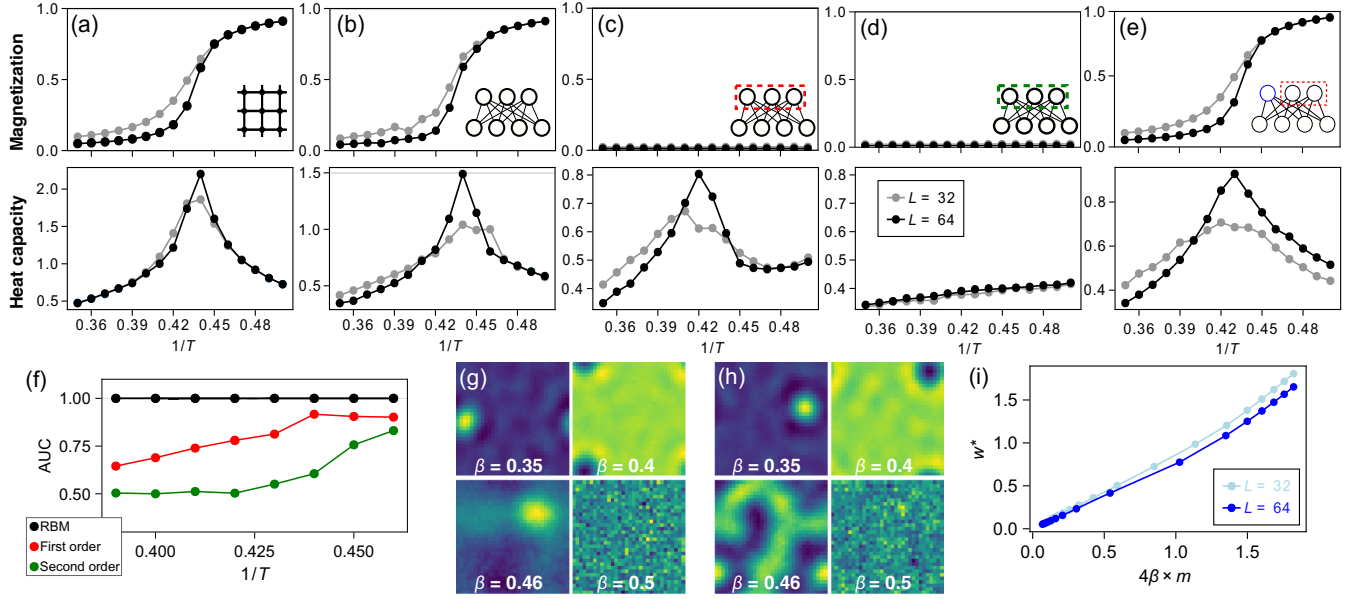


FIG. 7. Learning RBMs on two-dimensional Ising model data. (a) Magnetization and heat capacity as functions of the temperature for the samples generated by the Ising model (13). (b) Magnetization and heat capacity of samples generated by the RBM trained on the Ising data. (c) Magnetization and heat capacity of samples generated by the RBM with constraint (6) acting on all hidden units. (d) Magnetization and heat capacity of samples generated by the RBM with quadratic constraint (9) acting on all hidden units. (e) Magnetization and heat capacity of samples generated by the RBM with linear constraint (6) acting on all but one hidden unit. (f) Maximum AUC of classifiers trained to predict the sign of the sample magnetization from the RBM inputs. (g, h) Typical weights learned by the RBM at selected temperatures ($1/T = 0.35, 0.4, 0.46, 0.5$) for the unconstrained RBM and for the RBM with the first-order constraint. (i) Free weights attached to the released hidden unit compared to 4β times the magnetization of the Ising model.

where the sum runs over pairs (i, j) of nearest neighbors on a two-dimensional squared grid with $L \times L$ sites. Each spin v_i can take ± 1 values. We choose periodic boundary conditions; that is, site $(1, 1)$ is interacting with sites $(1, 2)$, $(2, 1)$, $(L, 1)$, and $(1, L)$. The model assigns probabilities given by the Boltzmann law $P_{\text{Ising}}(\mathbf{v}) \propto e^{-\beta E(\mathbf{v})}$ to configurations \mathbf{v} , where β is the inverse temperature; we hereafter denote the average over P by $\langle \cdot \rangle$. In the infinite- L limit, the model undergoes a phase transition from a paramagnetic phase ($\beta < \beta_c$) in which the magnetization

$$m = \left\langle \left| \frac{1}{N} \sum_i v_i \right| \right\rangle \quad (14)$$

vanishes, to a ferromagnetic phase ($\beta > \beta_c$) in which $m > 0$ [27]. The transition occurs at a critical inverse temperature $\beta_c \approx 0.44$ computed exactly by Onsager [36]; see Fig. 7.

A. Sampling the Ising model at equilibrium

We start by generating up to 10^6 samples from the Ising model through Monte Carlo (MC) simulations at different inverse temperatures in the range $0.35 \leq \beta \leq 0.5$. To quickly equilibrate at all temperatures, the MC chain includes both local Metropolis updates and global Wolff cluster moves known to be efficient to sample the model near β_c [37]; details about the implementation can be found

in Supplemental Material [26] Appendix A. The magnetization M and the heat capacity

$$C = \frac{\beta^2}{N} (\langle E^2 \rangle - \langle E \rangle^2) \quad (15)$$

are shown as functions of the inverse temperature in Fig. 7(a) for two system sizes $L = 32$ and $L = 64$. Additional observables such as the susceptibility

$$\chi = \frac{\beta}{N} \left[\left\langle \left(\sum_i v_i \right)^2 \right\rangle - \left\langle \left| \sum_i v_i \right| \right\rangle^2 \right] \quad (16)$$

and the correlation length are reported in Supplemental Material [26] Fig. S3. A peak in the heat capacity (and in the susceptibility) signals the crossover between the two phases, when β gets close to β_c , with a shift that vanishes with increasing L as predicted by finite-size-effects theory.

B. Learning with a standard RBM

We then use the MC samples as training data for an unconstrained RBM, with visible units taking ± 1 values. To enforce the global sign symmetry of the energy, i.e., $E(-\mathbf{v}) = E(\mathbf{v})$ [see Eq. (13)], we choose hidden units $h_\mu = \pm 1$ (instead of 0, 1 as in the MNIST case) and vanishing biases on the both visible ($g_i = 0$) and hidden ($\theta_\mu = 0$)

units. The training phase thus consists in inferring the RBM weights $w_{i\mu}$ only.

We verify that the log-likelihood $\log P(\mathbf{v})$ of test MC data estimated with the trained RBM correlate with the Ising energy $E(\mathbf{v})$ (Supplemental Material [26] Fig. S7). The weights learned by the RBM exhibit localization patterns [see Fig. 7(g)] at low temperatures, in agreement with observations reported in previous works on the one-dimensional Ising model [29].

We generate samples from these RBMs learned at different β 's using alternate Gibbs sampling and evaluate the magnetization, heat capacity, and susceptibility. The results are in agreement with the same quantities computed from samples of the Ising model distribution; see Fig. 7(b). This observation is consistent with literature [28,38,39], where RBMs were shown to be able to accurately fit statistical physics models such as the Ising model.

C. Partial erasure of information with a fully constrained RBM

We hereafter choose that the label $u = \pm 1$ associated with a configuration of spins \mathbf{v} is the sign of its magnetization,

$$u(\mathbf{v}) = \text{sign}\left(\sum_i v_i\right). \quad (17)$$

1. Linear constraints

By symmetry, the vector $\mathbf{q}^{(1)}$ in Eq. (6) has uniform components $q_i^{(1)} = q^{(1)}$ due to the translation invariance of the lattice resulting from periodic boundary conditions. Imposing the linear constraint in Eq. (6) thus amounts to demanding that all weight vectors sum up to zero, i.e., $\sum_i w_{i\mu} = 0$ for $\mu = 1, \dots, M$.

We then train a RBM on the MC data under these constraints. The log-likelihoods of test Ising configurations are poorly correlated with the Ising model energies in Eq. (13); see Supplemental Material [26] Fig. S8. In addition, RBM-generated samples show no magnetization at any inverse temperature, even for $\beta > \beta_c$; see Fig. 7(c). Surprisingly, however, other observables such as the heat capacity [Fig. 7(c)] or the susceptibility (Supplemental Material [26] Fig. S3) exhibit a peak at the crossover inverse temperature. We conclude that the constrained RBM-generated spin configurations with zero first moment, but a substantial part of higher-order correlations, is still correctly captured and reproduced. We come back to the interpretation of the effective energy corresponding to this fully constrained RBM in Sec. V E.

2. Quadratic constraints

We next apply second-order constraints (9) to all weight vectors of the RBM. Because of the global invariance of the

Ising energy under spin reversal, $\mathbf{q}^{(2)} = 0$ abiding to definition (10). However, the reversal symmetry is lifted in the presence of an arbitrary small uniform external field Δ , i.e., $E(\mathbf{v}) \rightarrow E(\mathbf{v}) - \Delta \sum_i v_i$. We show in Supplemental Material [26] Appendix G that, to first order in Δ , $\mathbf{q}^{(2)} \simeq \frac{1}{2} \Delta \mathbf{Q}^{(2)}$ with

$$Q_{i,j}^{(2)} = \left\langle \left| \sum_k v_k \right| v_i v_j \right\rangle_{\mathcal{D}} - \left\langle \left| \sum_k v_k \right| \right\rangle_{\mathcal{D}} \langle v_i v_j \rangle_{\mathcal{D}}. \quad (18)$$

The tensor $\mathbf{Q}^{(2)}$ can be estimated numerically and used to constrain the weight vectors through Eq. (9).

RBM's learned under these quadratic constraints generate spin configurations with zero magnetization, as in the case of linear constraints; see Fig. 7(e). Remarkably, the specific heat and the susceptibility show no peak as β is varied, suggesting that quadratic constraints on the weights have much stronger impact on the distribution of spin configurations. The heat capacity, in particular, has a mild monotonic increasing tendency with β , attaining similar values to the original model at low and high temperatures. However, inference of the magnetization sign is still possible from the hidden representation, although with degraded performance. For each inverse temperature, we train classifiers of varying complexity and measure their performance in predicting the labels. The resulting AUCs are shown in Fig. 7(f) and are above chance level (0.5) at high β . This indicates that higher-order correlations presumably present in the inputs of full-constrained RBMs (such as the Binder cumulant [40]) can be used for predicting labels with some success; we encounter a similar situation in the MNIST0/1 case.

D. Manipulating representations and spin configurations with a partially constrained RBM

We now apply constraint (6) on all but one hidden unit when training the RBM on the Ising data. The released hidden unit, hereafter referred to as h^* , learns a weight vector which is approximately proportional to $\mathbf{q}^{(1)}$; that is, the weights connecting to h^* are uniform over the visible layer, with a common value hereafter referred to as w^* . The resulting RBM then has one hidden unit that controls the sign of the magnetization of the generated samples, while the remaining hidden units capture local correlated patterns of neighboring spins. Indeed, the constrained weights display localized patterns similar to those of an unconstrained RBM [Fig. 7(e)]. In addition, the RBM reproduces the behavior of all observables as the inverse temperature is varied [Fig. 7(e) and Supplemental Material [26] Fig. S3]. These results strongly suggest that the constraints on (all but one) weight vector applied during learning do not impair the ability to fit the data but serve only to reorganize the latent representations. In addition to Eq. (6), we can also

impose constraints (9) on all but one hidden unit, with similar results to those reported (not shown).

E. Effective energy resulting from constraints

A heuristic argument allows us to better understand the nature of the distribution expressed by the fully constrained RBM (linear case), in particular, why generated configurations have zero magnetization while encoding nontrivial spin-spin correlations [Fig. 7(c)].

Let us first notice that the general expression for the log-probability of a visible configuration \mathbf{v} in the RBM reads, due to the absence of biases on the units,

$$\log P_{\text{RBM}}(\mathbf{v}) = \sum_{\mu=1}^M \log \cosh \left(\sum_i w_{i\mu} v_i \right), \quad (19)$$

up to an irrelevant additive constant. This formula applies in particular to the released RBM of Sec. V D, in which all but one hidden unit, say, $\mu = 1$, are constrained to satisfy Eq. (6). Based on our previous finding that $w_{i,1} \simeq w^*$, we obtain

$$\log P_{\text{rel}}(\mathbf{v}) \simeq \sum_{\mu=2}^M \log \cosh \left(\sum_i w_{i\mu} v_i \right) + w^* \left| \sum_i v_i \right|, \quad (20)$$

where we approximate $\log \cosh x \simeq |x|$ for large arguments x and again neglect additive constants. Based on Eq. (20), we may proceed in two steps. First, as we empirically find that the released RBM is a good approximation of the ground-truth Ising distribution, we approximate $\log P_{\text{rel}}$ with $\log P_{\text{Ising}}$. Second, the first term on the right-hand side of Eq. (20) expresses the log-probability of \mathbf{v} computed by a RBM with weight vectors constrained to be orthogonal to $\mathbf{q}^{(1)}$ and can thus be identified with $\log P_{\text{constr}}$. We conclude, using Eq. (13), that the effective energy function on the spin configuration encoded by the fully constrained RBM is approximately equal to

$$E_{\text{constr}}(\mathbf{v}) \simeq - \sum_{(ij)} v_i v_j + \frac{w^*}{\beta} \left| \sum_i v_i \right|. \quad (21)$$

The effect of the constraints on the weights is to introduce an L_1 -like penalty against magnetized configurations opposing the Ising energy, which tends to align spins. This explains both the disappearance of magnetization and the remanent correlations observed in Fig. 7(c).

We can also estimate the value of w^* selected through learning of the fully constrained RBM, with a heuristic argument. Consider a typical configuration of the Ising model at low temperature, i.e., in the ferromagnetic regime corresponding to magnetization $m^* \neq 0$. The effective field acting on spin, say, i , reads, according to Eq. (21),

$$g_i^{\text{eff}} = \sum_{j \in \mathcal{N}_i} v_j - \frac{w^*}{\beta} \text{sign}(m^*), \quad (22)$$

where \mathcal{N}_i refers to the neighborhood of spin i on the squared grid. Taking the average over the spin i , we obtain the mean value of the effective field

$$\langle g^{\text{eff}} \rangle = z m^* - \frac{w^*}{\beta} \text{sign}(m^*), \quad (23)$$

where $z = 4$ is the coordination number on the grid. We conclude that the effective field vanishes when

$$w^* = \beta z |m^*|. \quad (24)$$

The above expression gives the minimal strength of the L_1 penalty capable of counterbalancing the local interactions tending to magnetize spins. It is expected to vanish in the paramagnetic regime. Higher values are disfavored during the RBM training phase, as they would assign higher energies E_{constr} in Eq. (21) to typical magnetized Ising configurations, and thus lower likelihoods.

We compare the heuristic estimate for w^* provided by Eq. (24) to the numerical results for w^* obtained from training partially constrained RBM on 2D Ising data in Fig. 7(i). Despite the presence of finite-size effects, we observe a good agreement between Eq. (24) and the simulation results.

VI. APPLICATION TO MNIST HANDWRITTEN DIGIT IMAGES

We next consider the MNIST handwritten digit dataset [24]. Pixel intensities are binarized by thresholding at 0.5. For simplicity, we start by considering the subset of images containing only digits 0 and 1 (MNIST0/1), for which the class label u is binary.

A. Learning with a standard RBM

We train a standard RBM on MNIST0/1, with $M = 400$ binary hidden units and $N = 28 \times 28$ visible units, through maximization of the log-likelihood (4) (see Supplemental Material [26] Appendix A 6 for further details). Figure 8(a) shows Markov chains of samples derived from Gibbs sampling of the resulting models. The machine generates strings of 0s or 1s, depending on the initial condition, with very rare transitions between these classes. Note that the absence of transitions from 0 to 1 (or vice versa) is likely due to the strong dissimilarities between these two digits in configuration space and the lack of low-energy configurations connecting them; training the RBM on all digits tends to connect these two modes and increase the frequency of observed transitions.

To quantify the information content in the inputs about the labels (digit value), we estimate the mutual information

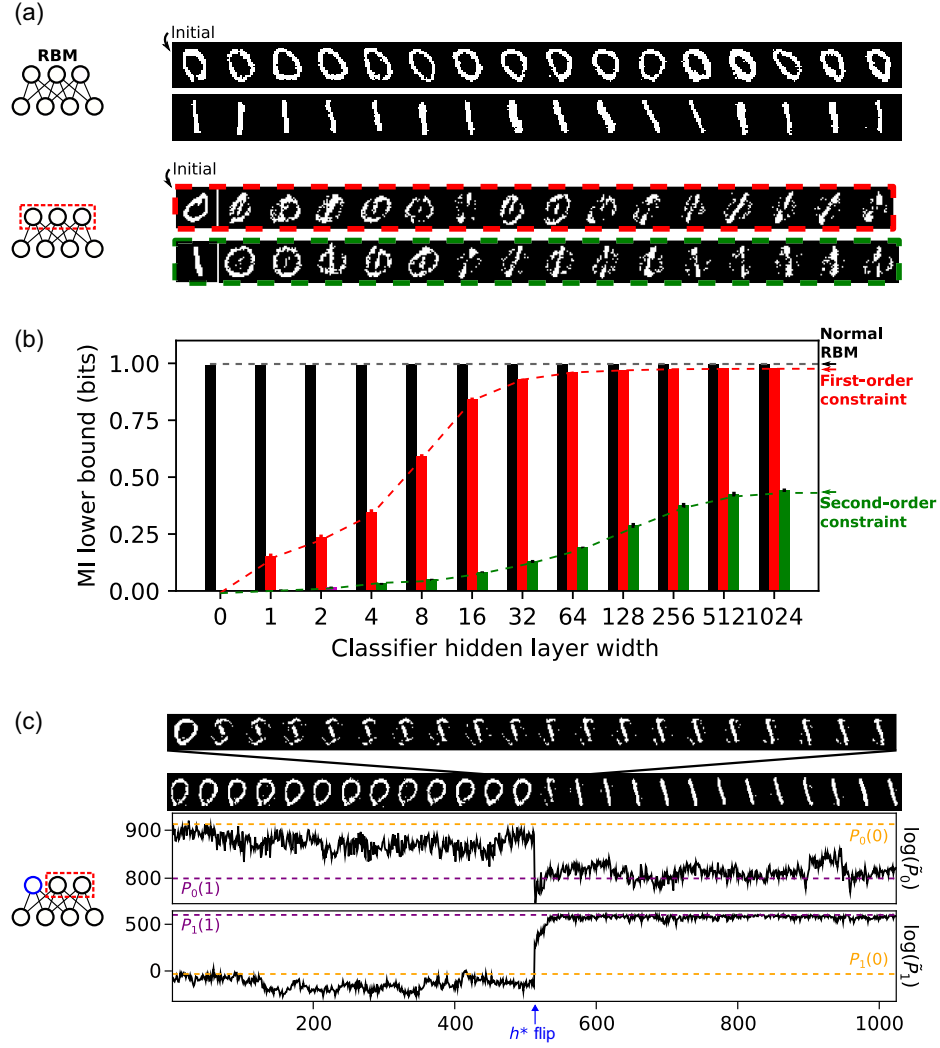


FIG. 8. Manipulating representations of the RBM trained on MNIST0/1. (a) Samples generated by the RBM initialized with a data image (0 or 1). Top two rows show a standard (unconstrained) RBM. Bottom two rows show samples from the RBM trained with linear (red dashed) and quadratic (green dashed) constraints. In both cases, a Markov chain is generated by Gibbs sampling (starting from a 0 or 1 data digit), and images are saved every 64 steps, until reaching a total of 16 samples as shown. (b) Lower bound $\mathcal{S}_{\text{label}} + \mathcal{L}_{\text{class}}$ to the mutual information between inputs and labels [see Eq. (25)] vs classifier width. The bounds to MI are measured in bits and shown in discontinuous lines. Colors correspond to the different RBM models. Black: standard and unconstrained. Red: fully constrained with linear constraints; see Eq. (6). Green: fully constrained with quadratic constraints; see Eq. (9). (c) Samples from the RBM trained with first-order constraint acting on all but one hidden unit, which is flipped at the middle of the MC chain (blue arrow). Starting from a 0 data digit, samples are saved every 64 Gibbs steps. Top panel shows an enlarged view of the transition, with images every three steps instead. The lower panels show the logarithm of the unnormalized probability $\ln \tilde{P}(\mathbf{v}) = \ln (\sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})})$ of generated digits by constrained RBMs, evaluated on RBMs trained only on 0s (RBM0) or 1s (RBM1). Purple and orange dashed lines correspond to the average $\ln \tilde{P}(\mathbf{v})$ of data digits 0 and 1.

$\text{MI}(u, \mathbf{I}(\mathbf{v}))$. While computing MI is very hard, a tractable lower bound can be obtained through the Gibbs variational inequality [30],

$$\begin{aligned} \text{MI}(u, \mathbf{I}(\mathbf{v})) &\geq \sum_{u, \mathbf{v}} P_{\mathcal{D}}(u, \mathbf{v}) \ln \left(\frac{P_{\text{class}}(u | \mathbf{I}(\mathbf{v}))}{P_{\mathcal{D}}(u)} \right) \\ &= \mathcal{S}_{\text{label}} + \mathcal{L}_{\text{class}}, \end{aligned} \quad (25)$$

where $P_{\mathcal{D}}(u, \mathbf{v})$ is the empirical distribution of labeled data, and $P_{\text{class}}(u | \mathbf{I}(\mathbf{v}))$ is any conditional distribution implemented here by a classifier attempting to predict the label. By rearranging terms, this equals the entropy of labels in the data ($\mathcal{S}_{\text{label}}$) plus the log-likelihood of the classifier averaged over held-out data ($\mathcal{L}_{\text{class}}$).

This lower bound to MI is shown in Fig. 8(b) (black bars) for classifiers of increasing complexity corresponding to two-layer networks with a hidden layer of increasing

width (horizontal axis in the figure); see Supplemental Material [26] Appendix E for details about the architecture and training of these classifiers. The simplest network is a linear classifier (perceptron, width = 0), and already achieves nearly perfect prediction accuracy. In addition, the weights of this optimal linear classifier are distributed over all hidden units, showing that information about the label is distributed across the latent representation. As the width of the classifier increases, the lower bound to MI saturates at a value close to 1 bit, the maximum possible for two label classes, indicating that the RBM inputs capture maximum label information. We emphasize that the RBM has no direct access to the label values during training.

B. Partial erasure of information with a fully constrained RBM

We next train a RBM with a constraint applied on the weight vectors attached to all hidden units.

1. Linear constraints

Figure 8(a) (bottom, red) shows typical configurations generated by a RBM trained with constraints (6). As expected, these configurations tend to be blurred mixtures of 0s and 1s.

A simple linear discriminator looking at the inputs to the hidden units is unable to predict the labels of these digits, in agreement with the adversarial interpretation of Eq. (6). However, information about the digit class is still present in the RBM representations through higher-order correlations. Sufficiently complex classifiers are able to recover the label of data digits with maximum accuracy [Fig. 8(b)] and give lower bounds to MI close to unity. This result shows that, while condition (6) is not sufficient to erase the label information from the representation extracted by the RBM, it does make retrieval of this information more difficult.

2. Quadratic constraints

Imposing the stronger, quadratic constraints in Eq. (9) results in a sample of worse quality; see green row in Fig. 8(a), bottom. Figure 8(b) shows that simple classifiers trained are unable to predict the labels from the inputs. Interestingly, more complex classifiers achieve a moderate nonzero prediction accuracy but provide substantially lower estimates of the mutual information than when trained on linearly constrained RBMs (compare green and red bars). The lower bounds to MI seem to saturate to a value well below 1 as the classifier widths increase. These results indicate that quadratic constraints erase a sizable part of the information about the labels.

3. On the generative power of the fully constrained RBM

Configurations sampled from the fully constrained RBMs in Fig. 8(a) (bottom) tend to be blurred mixtures of digits (0 and 1). In this case, the data are in fact a mixture

of two widely separated distributions associated with 0s and 1s. This is reminiscent of configurations of opposite magnetization in the Ising model at low temperature in Sec. V D, and the sampled blurred digits are in analogy to the “intermediate” configurations of zero magnetization that the fully constrained RBM samples in that case [Fig. 7(c) top]. We however see that in the Ising model, the configurations sampled from the fully constrained RBM still carry relevant information in higher-order statistics, e.g., as shown by the behavior of the heat capacity; see Fig. 7(c) bottom.

To illustrate how fully constrained a RBM can generate samples with meaningful information present in higher-order statistics in the setting of handwritten digit images, we consider the following simple numerical experiment. For each 0 digit from MNIST, we produce an additional image where pixel colors are flipped (producing black zeros in white background) and define a binary label encoding the background color. We then train a fully constrained RBM on these data. Generated samples are shown in Supplemental Material [26] Fig. S6. The fully constrained RBM generates recognizable 0 digits embedded in noisy backgrounds, where local patches in the digit strokes clearly tend to share the same color, indicating that the overall structure of the digit is preserved through correlations.

C. Manipulating representations and digits with a partially constrained RBM

We now impose linear constraints (6) to all but one (blue) hidden unit. As we state in objective B, our intention is to promote concentration of label information on this released unit; see Fig. 3(b). After learning, the released weight vector is similar (up to a global scale factor) to vector $\mathbf{q}^{(1)}$ (Supplemental Material [26] Fig. S5), a direction forbidden to the other hidden units. Hence, the average value of the unit conditioned to a visible configuration (digit) is an excellent predictor of the corresponding label.

Samples generated by the RBM are nice-looking 0s or 1s, in a manner consistent with the state of the released hidden unit. Furthermore, manipulating the state of this hidden unit, i.e., freezing it to 0 or 1, helps generate samples with the desired labels. We show in Fig. 8(d) numerical experiments illustrating the effects of such manipulations. We initialize the RBM with a digit [0 in Fig. 8(d)] extracted from the MNIST0/1 dataset, and sample new configurations through alternate Gibbs samplings. As with standard RBM, the samples vary over time, but the digit class remains unchanged. We then flip the state of the hidden unit [middle of Fig. 8(d)]. As a consequence, the resulting visible configuration converges to the other digit class after some short transient (see top part of panel).

To evaluate the quality of the generated digits, we train two RBMs only on 0s or 1s, respectively, and evaluate the log-likelihoods of the generated digits on two standard RBMs, one trained with 0 digits only, and another trained

on 1s only. These two machines provide expected reference scores for 0s and 1s. Figure 8(e) shows that the generated digits are of good quality, with log-likelihood values comparable to the ones of the data.

D. Case of more than two digits

While we have focused on the case of binary labels so far, our approach can be easily adapted to more than two classes. We consider the case of D classes and use one-hot encoding for the labels; i.e., we introduce D labels u_d , one for each class $d = 0, 1, \dots, D-1$. Because of the one-hot encoding prescription, each data configuration \mathbf{v} is such that $D-1$ labels $u_d(\mathbf{v})$ vanish and one is equal to 1.

Analogous to Eq. (6), we define D vectors (in the N -dimensional space of data)

$$\mathbf{q}_d^{(1)} = \langle u_d(\mathbf{v})\mathbf{v} \rangle_{\mathcal{D}} - \langle u_d(\mathbf{v}) \rangle_{\mathcal{D}} \langle \mathbf{v} \rangle_{\mathcal{D}}. \quad (26)$$

We then generalize Eq. (6) to multiple classes by imposing that weight vectors be orthogonal to all $\mathbf{q}_d^{(1)}$, with $d = 1, \dots, D$. It is easy to check that the D vectors in Eq. (26) sum up to zero, a consequence of the one-hot encoding scheme. We therefore consider only the last $D-1$ vectors, with indices $d = 1, 2, \dots, D-1$ to obtain linearly independent constraints acting on the weights.

In practice, the constraints $\mathbf{w}^\mu \perp \mathbf{q}_d^{(1)}$ are enforced through the architecture shown in Fig. 9(a) in which a set of $D-1$ hidden units h_d are released, each with respect to a single $\mathbf{q}_d^{(1)}$ and constrained to be orthogonal to all the other $D-1$ vectors. In this way, when activating one of these hidden units, say, μ , the corresponding digit $d = \mu$ is expected to be sampled on the visible layer. When all first $D-1$ hidden units are silent, digit $d = 0$ is expected to be sampled.

We illustrate this approach in the case of $D = 4$ digits, with RBMs trained from MNIST0/1/2/3. The vectors $\mathbf{q}_d^{(1)}$ in Eq. (26) are shown in Fig. 9(b). After training the RBM under the orthogonality constraints, the released hidden units $\mu = 1, 2, 3$ are strongly activated by, respectively, digits $d = 1, 2, 3$. In Fig. 9(c), we show the average inputs to these hidden units when data digits are presented on the visible layer of the RBM; the corresponding weight vectors are depicted in Fig. 9(d). When digit 0 is present on the visible layer, the three hidden units are silent. Other hidden units are weakly activated by the different digits and capture information (small stretches, local contrast) crucial for generating high-quality digits but not directly related to their identity; see panel “other” in Fig. 9(c).

We next manipulate these units to generate digits out of one of the four classes. The outcome is shown in Fig. 9(e), where the Markov chain is initialized with a 1 digit from the MNIST data, and the first released hidden unit ($\mu = 1$) is on, while the other two ($\mu = 2, 3$) are off. Sampling the RBM in this condition generates a string of 1s as illustrated in the figure. Turning this unit off and turning the second

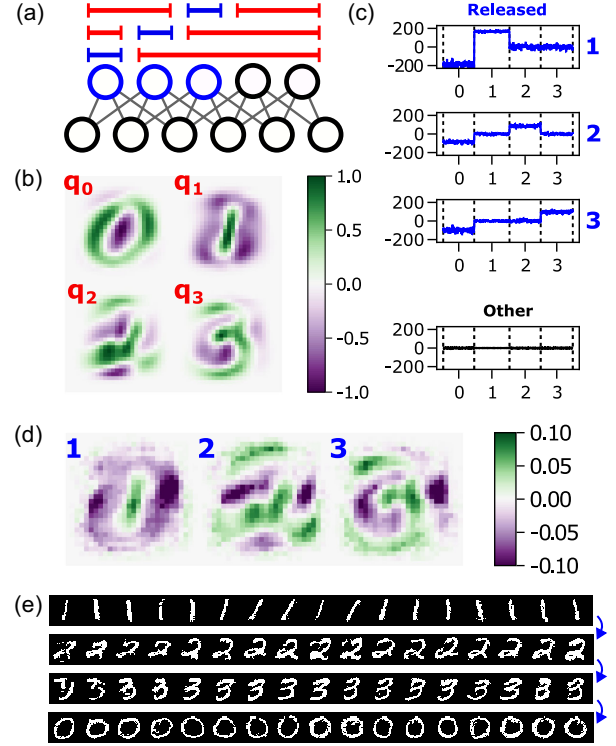


FIG. 9. Manipulating representations of a RBM trained on MNIST0/1/2/3. (a) Sketch of the constraints applied to hidden-unit weights in the case of multiple classes, here, $D = 4$. (b) Vectors $\mathbf{q}_d^{(1)}$ for digit classes 0, 1, 2, and 3; see Eq. (26). (c) Inputs received by the three released hidden units [in blue on panel (a)], when the 6000 digit images in classes 0, 1, 2, and 3 are presented (x axis). In the fourth, bottom panel, the inputs received by a random hidden unit from the constrained group (black) are shown. (d) Weights $w_{1\mu}$ learned by the released hidden units $\mu = 1, 2, 3$. (e) Samples generated from this machine by Gibbs sampling (images shown are taken every 64 steps). The first (top row) released unit 1 is active, while the other two are inactive. Then, we activate unit 2 (second row) while inactivating unit 1 (blue arrow), and similarly for 3 (third row). In the last row, all three units are inactive.

$\mu = 2$ on now produces a transition in the visible layer and generates digits 2. Iterating this procedure, we generate 3s, and finally 0s by turning off all released hidden units [last row in Fig. 9(e)].

VII. APPLICATION TO PROTEIN SEQUENCES WITH TAXONOMY ANNOTATIONS

A protein family is a group of proteins that share a common evolutionary origin reflected by their related functions and similarities in sequence or structure [25]. Protein families are often arranged into hierarchies, with proteins that share a common ancestor subdivided into smaller, more closely related groups. In recent years, RBMs have been successfully applied to extract structural, functional, and evolutionary information from the sequences

attached to a protein family [19,20,41]. Our aim here is to use a partially constrained RBM to disentangle the label defining the taxonomic domain (eukaryota or bacteria) a protein sequence belongs to and manipulate the domain-determining hidden unit to drive a continuous transition, or morphing, between one taxonomic domain to the other during sampling of artificial sequences.

A. The *K*-homology domain

To illustrate the application of our model, we select the KH module, a common nucleic-acid binding motif in proteins found in multiple species, both eukaryotic and prokaryotic. Structurally, KH domains adopt a globular fold constituted by three alpha helices and three beta sheets [42–44], as shown in Fig. 10(a). A central feature of the KH domain is the presence of a signature Iso-Gly-X-X-Gly motif [see Figs. 10(a) and 10(b)] conserved across the entire family, which in cooperation with flanking helices, forms a cleft

where recognition of four nucleotides in single-stranded DNA or ribonucleic-acid chains occurs [44]. Mutations in these highly conserved residues result in loss of function [45]. In particular, substitution of the moderately conserved isoleucine following the Gly–Gly loop (two sites after) by Asn, in a KH domain locus of the fragile-X mental retardation gene in humans causes fragile-X syndrome, a leading heritable cause of mental retardation [46].

We select this family in our work as it has a sufficient number of eukaryotic and bacterial sequences available in the Pfam database [25]. The PF00013 family of homologous sequences includes approximately 11 000 bacterial sequences and approximately 38 000 eukaryotic sequences of the KH domain. After aligning, removing insertions, and retaining only columns with less than 50% gap (deletions) content, the sequences end up having a common length of $L = 62$ amino acids. As the taxonomic origin of every sequence can be simply queried through the Uniprot database [48], we define label $u = 0$ and 1 for, respectively,

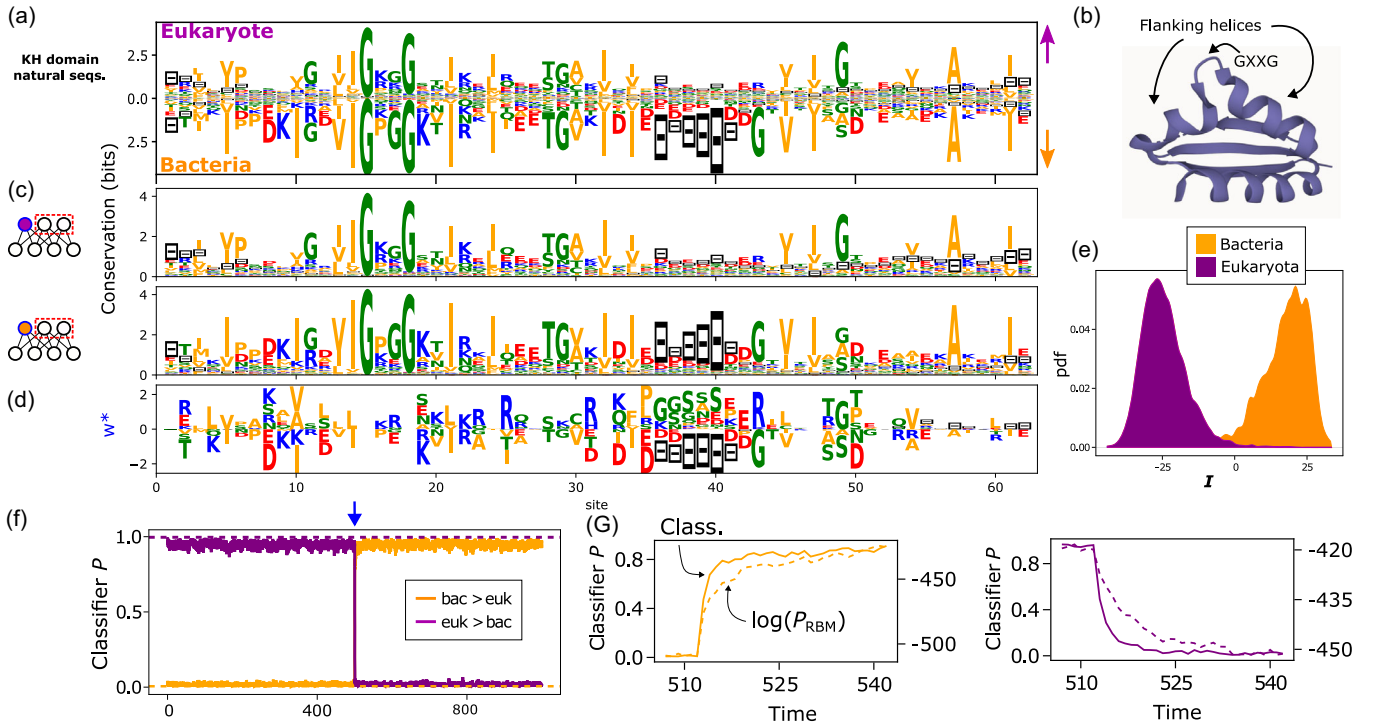


FIG. 10. Taxonomy of protein families. (a) Sequence logos of eukaryotic (purple, above) and bacterial (orange, below) sequences from the PF00013 protein family. We use the following color code: green for polar residues, blue for basic, red for acidic, and orange for hydrophobic. Gaps are shown in black. (b) Ribbon structure of KH domain showing locations of the Gly–Gly loop and flanking helices. Image prepared with Mol* Viewer [47]. (c) Sequence logos of 100 000 generated sequences when the released hidden unit is set to 1 (top) or 0 (bottom). To ensure that sampling is equilibrated, we track the average and standard deviation of the energy of the samples in time and see that these statistics are essentially constant after approximately 200 steps, suggesting that samples can be collected every 5000 steps. (d) Weights of the released hidden unit. (e) Inputs received by the released hidden unit when presented with sequences from the two classes. (f) Markov chain started from bacterial (orange) or eukaryotic (purple) sequences from the data. The panel shows the probability of being a eukaryotic vs bacterial sequence in a perceptron classifier. Discontinuous lines are the average value for the data sequences of each class. A total of 1024 Gibbs sampling steps are taken, and the flip of h^* occurs at step 512 (blue arrow). (g) Enlarged view near the transition, showing also the log of the unnormalized marginal $[\log \tilde{P}_{\text{RBM}}(\mathbf{v})]$ of sampled sequences (right axis) evaluated on a RBM trained on the full family.

bacterial and eukaryotic proteins. To reduce common ancestry bias, the sequences are weighted according to their dissimilarity to other members of the same family [49,50]: The weight assigned to a sequence is proportional to the inverse of the number of sequences in the family with a Hamming distance smaller than 20% of the sequence length. We also balance the total weights of eukaryotic and bacterial classes so that both classes have equal weights.

Figure 10(a) shows the sequence logos of the eukaryotic (top) and bacterial (bottom) sequences in the family after carrying out the above preprocessing steps. Some features are shared across KH domain sequences in both subfamilies, such as the well-conserved Gly–Gly loop [Fig. 10(b)]. Bacterial sequences have an overall larger content of gaps (deletions) with respect to the consensus alignment, reflecting sequence length differences in the two subfamilies.

B. Learning a generative model with a standard RBM

Multiple sequence alignments are represented using categorical or Potts variables, each site of the alignment having one of 21 possible values (20 amino acids and one gap value). Gaps are necessary to model sequences of varying lengths [49]. Using the one-hot encoding, a configuration \mathbf{v} of the visible layer encodes a sequence over $21 \times L$ units, where L is the sequence length.

We first train a RBM on the full alignment containing both eukaryotic and bacterial sequences, following Ref. [19]. The RBM captures statistics of the sequence alignment, such as conservation profiles at each site. In addition, simple linear classifiers trained on top of the hidden layer of the RBM achieve AUCs of 0.9 in distinguishing between these two classes.

C. Fully constrained RBMs are still able to generate foldable sequences

We then train a RBM with constraint (6) acting on all hidden units. The resulting model continues to match the conservation profile of the MSA and generates diverse sequences. We furthermore validate the foldability of sampled sequences using ALPHAFOLD [51]. As explained in Supplemental Material [26] Appendix F, we compute the template-matching score of the predicted structures of sampled sequences in comparison to the natural sequences, obtaining values > 0.7 for both the standard RBM and the fully constrained RBM, suggesting that these sequences are able to adopt the expected three-dimensional fold of the family. This result is in agreement with objective A: The model distribution should preserve all the data features unrelated to the label.

D. Changing taxonomic domain with protein design

We then apply the linear orthogonality constraint in Eq. (6) to all but one weight vector. The weights of the

released hidden unit after training are shown in Fig. 10(d) and capture features that differentiate the two classes. For example, bacterial sequences tend to have deletions (gaps) around positions 35–40 of the alignment, indicating that this segment is often absent in bacterial sequences. The learned w_i^* reflect this difference by assigning negative weights to the gap symbol in this region. As a consequence, the distribution of inputs subtended by eukaryotic and bacterial sequences is well separated on this unit [Fig. 10(e)]. Conversely, features shared by eukaryotes and bacteria, such as the Gly–Gly loop, or the conserved I22, are ignored by \mathbf{w}^* .

We generate many samples from the RBM distribution, each conditioned to a fixed state of h^* , corresponding either to bacterial ($h^* = 0$) or eukaryotic ($h^* = 1$) classes. The sequence logos of the two sets of generated sequences are shown in Fig. 10(c); they closely match the ones of the training data. The list of differences between the logos associated with the two sequence domains include the following:

- (1) The Gly–Gly loop is followed by a conserved Lys19 predominantly in bacteria, but not so in eukaryotic sequences.
- (2) Bacterial sequences conserve a Asp–Lys–Iso motif (positions 8–10), which the RBM with $h^* = 0$ correctly emits, but not so in the $h^* = 1$ case.
- (3) Besides the two Gly conserved in the entire family, eukaryotic sequences also conserve Gly49, a site which appears less conserved in bacteria which admit also Ala or Ser at this position. The RBM correctly observes these variations.
- (4) Iso10 is highly conserved in bacteria, while in eukaryotes this site is not conserved, admitting, in particular, Val, Ala.

These examples suggest that the RBM can sample each subfamily, conditioned on the value of h^* .

Next, we sample the RBM starting from one bacterial or one eukaryotic sequence in the dataset as the initial condition, and with h^* set to the value matching the initial condition. After some steps, the value of h^* is flipped, and we monitor the dynamical evolution of the generated samples. Figure 10(f) shows the probability that generated sequences are eukaryotic or bacterial, according to a linear classifier achieving AUC > 0.9 on held-out test data (see Supplemental Material [26] Fig. S4).

Figure 10(g) shows a magnified view of the classifier probabilities and the log-likelihood in the vicinity of the hidden-unit switch. We evaluate the log-likelihood of the samples with a RBM trained on the full family (denoted $\log \tilde{P}_{\text{RBM}}$ in the figure). The class switch, as measured by the classifier score, occurs faster than the relaxation dynamics following the h^* flip, as measured by the likelihood. This suggests that the sampled sequences retain other features unrelated to the labeled class that relax at a slower rate.

VIII. ROBUSTNESS AGAINST THE SCARCITY OF LABELED DATA

One important advantage of our approach is that labeled data are only necessary to estimate the vector $\mathbf{q}^{(1)}$ (7) used in the first-order constraint (6), or the matrix $\mathbf{q}^{(2)}$ (10) in the case of the second-order constraint (9). Having determined $\mathbf{q}^{(1)}$ or $\mathbf{q}^{(2)}$, the training of the RBM benefits from additional unlabeled data, and in this regard, our model is semisupervised. This property is useful in many real applications, where labels are assigned by humans, are costly to obtain, and thus available for only a small fraction of the data. An example is the KH domain protein-sequence dataset considered in Sec. VII, where we are able to collect reliable taxonomic labels for only 10% of the sequences.

To better understand the amount of labeled data needed for our approach to be effective, we conduct further numerical experiments in which the fraction of labeled data is progressively decreased. We consider below the linear constraint and the MNIST0/1 data for the sake of simplicity. Similar results for the KH domain are reported in Supplemental Material [26] Fig. S10.

Since $\mathbf{q}^{(1)}$ becomes trivially zero when there are no data in one of the label classes, we consider balanced subsampled labeled datasets with equal numbers of labeled examples in each class. Figure 11(a) shows the average overlap between vector $\mathbf{q}^{(1)}$ computed on such a subsampled labeled dataset (referred to as $\mathbf{q}_{\text{sub}}^{(1)}$), and the vector $\mathbf{q}^{(1)}$ computed on the full labeled dataset (denoted by $\mathbf{q}_{\text{full}}^{(1)}$), as a function of the number B of labeled examples available, divided by the dimension of the data N . Here, the overlap is defined by

$$\phi = \frac{\mathbf{q}_{\text{full}}^{(1)} \cdot \mathbf{q}_{\text{sub}}^{(1)}}{|\mathbf{q}_{\text{full}}^{(1)}| |\mathbf{q}_{\text{sub}}^{(1)}|}. \quad (27)$$

For each given number of labeled examples, we consider 100 random realizations of the subsampled labeled dataset and estimate the average of ϕ over these realizations. It can be seen from Fig. 11(a) that the overlap never drops below approximately 0.6. This result can be understood by considering the separation between the two classes of data (see inset in the figure). Writing the covariance matrix conditioned on the class label

$$C_{ij}^{(u)} = \langle v_i v_j | u \rangle - \langle v_i | u \rangle \langle v_j | u \rangle, \quad (28)$$

as well as the mean data vector associated with each class

$$v_i^{(u)} = \langle v_i | u \rangle, \quad (29)$$

we can derive a simple estimate connected to the average separation between the classes, $\mathbf{v}^{(0)} - \mathbf{v}^{(1)}$ and the variances inside each class $\text{Tr}C^{(0)}$, $\text{Tr}C^{(1)}$ (see Supplemental Material [26] Appendix H for a derivation) that writes

$$\langle \phi \rangle \approx \left(1 + \frac{1}{B} \frac{\text{Tr}(C^{(0)} + C^{(1)})}{\|\mathbf{v}^{(0)} - \mathbf{v}^{(1)}\|^2} \right)^{-1/2}, \quad (30)$$

where B is the total number of labeled examples, and the average is taken over all labeled datasets with $B/2$ examples in each class. Thus, the overlap increases with the separation between the classes ($\mathbf{v}^{(0)} - \mathbf{v}^{(1)}$) and decreases if the classes have large variances ($\text{Tr}C^{(0)}$, $\text{Tr}C^{(1)}$), as depicted in the inset

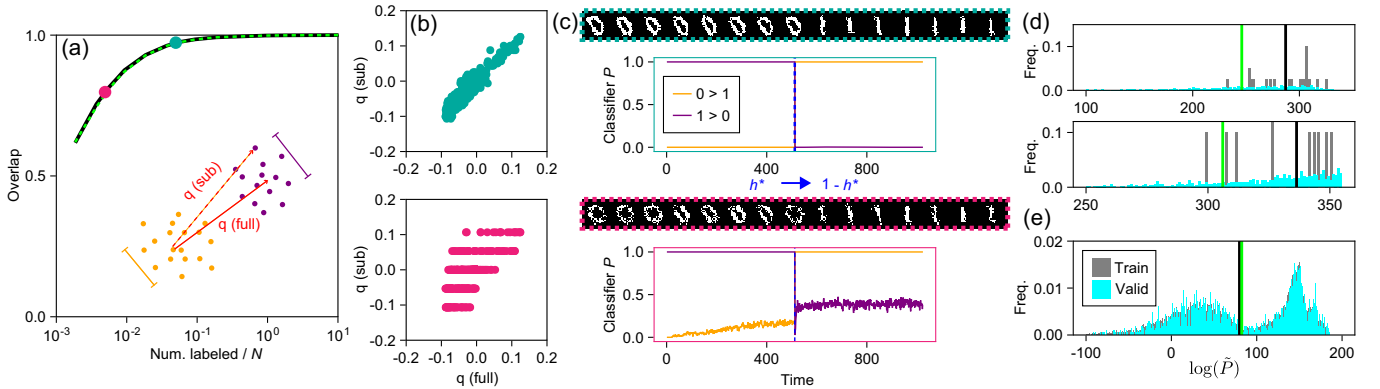


FIG. 11. Semisupervised training with a subsampled labeled dataset. (a) Overlap (27) between $\mathbf{q}_{\text{sub}}^{(1)}$ (computed on a subsampled-labeled dataset) and $\mathbf{q}_{\text{full}}^{(1)}$ (computed on the full dataset) plotted as a function of the number of labeled examples in the subsampled dataset divided by the dimension ($28 \times 28 = 784$ for MNIST). An average of over 100 random realizations of the subsampled dataset is taken. The black solid curve shows the empirical result, while the dashed green curve is the theoretical estimate (30). Inset shows a diagram of how class separation relates to the overlap in connection to (30). (b) For the pink and cyan dots of (a), we plot an example of the obtained vectors $\mathbf{q}_{\text{sub}}^{(1)}$ in comparison to $\mathbf{q}_{\text{full}}^{(1)}$. (c) Label manipulation using the subsampled $\mathbf{q}_{\text{sub}}^{(1)}$ in the two cases. (d) Histogram of log-likelihoods of (subsampled) training and withheld dataset for a RBM trained on a subset of 0 (top) or 1 (bottom) digits, corresponding to the labeled datasets used in the cyan dot in the previous panels. The black and green vertical lines indicate the average values. (e) Histogram of log-likelihoods of training and withheld dataset of the partially constrained RBM in the cyan setting of the previous panels.

of Fig. 11(a). The estimate (30) is plotted in Fig. 11(a) and is in excellent agreement with the empirical average overlap.

Figure 11(b) shows the scatter plots of the components of two example vectors $\mathbf{q}_{\text{sub}}^{(1)}$ computed from subsampled labeled data at the pink and cyan points highlighted in Fig. 11(a) vs the components of the vector $\mathbf{q}_{\text{full}}^{(1)}$ computed from all labeled data. Using these vectors $\mathbf{q}_{\text{sub}}^{(1)}$, we then train two RBMs subject to Eq. (6) acting on all but one hidden unit. Then we attempt to manipulate the sampled data by controlling this released hidden unit. The results are shown in Fig. 11(c). In both cases, the RBMs generate acceptable data and the state of the released hidden unit h^* correlates with the sampled digit, even though for the extremely subsampled case (pink) the digits tend to be more noisy.

To further underline the advantage of our method with respect to supervised learning in a situation with few labeled data, we train normal RBMs on the subsampled labeled data, specializing on 0 or 1 digits only. As expected for the small amount of training data, these models tend to overfit. This is shown in the histograms of log-likelihood assigned to training and a withheld validation dataset in Fig. 11(d) (top for 0 digits and bottom for 1s). The gap in the average log-likelihood of training and validation data (black and green vertical lines, respectively) is quite large, in both cases, indicating overfitting. In contrast, the partially constrained RBM (the same from the cyan dot in the previous panels of the figure) uses both the few labeled data and the large quantity of unlabeled data to

avoid overfitting, and we show the log-likelihood histograms for training and validation data in Fig. 11(e). The agreement between both subsets is excellent, indicating that this model is not overfitting.

In summary, these results provide evidence for the fact that our method is also applicable with limited labeled data.

IX. ESTIMATING THE COSTS OF PARTIAL ERASURE AND DISENTANGLEMENT

In this section, we estimate the cost associated with disentanglement (see Sec. III B 2), focusing on the impact of linear constraints on the weights. We resort to both numerical and analytical methods to estimate these costs.

A. Numerical estimates

Computing the likelihood requires estimating the normalization constant Z in Eq. (2). Since the exact calculation of Z is intractable, we use the annealed importance sampling (AIS) algorithm [52]. AIS estimates Z through a number of intermediate “annealed” distributions interpolating between the original RBM distribution and a simpler independent model that can be exactly sampled. This procedure provides a stochastic upper bound on the likelihood, which converges to the true value as the number of interpolating distribution increases. A stochastic lower bound can be obtained by a reverse interpolation procedure [53], which gradually “melts” the RBM back into the independent model; see Supplemental Material [26] Appendix A for details. Combining the

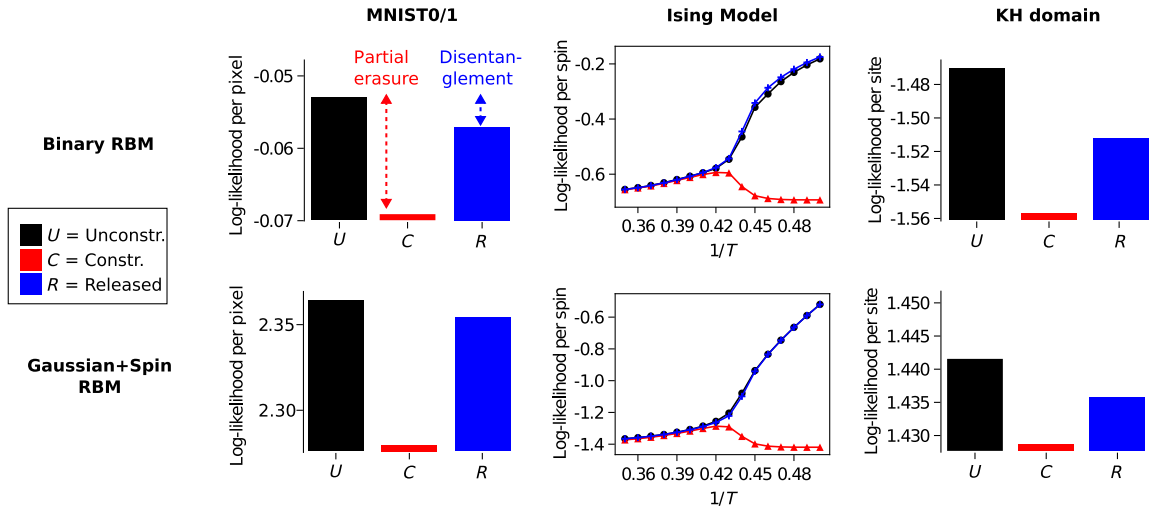


FIG. 12. Likelihood calculations. First row shows numerical estimates of the log-likelihood using RBMs with binary hidden units, along with the costs of applying Eq. (6) partially or on the full hidden layer. Bottom row shows analytical results obtained in a RBM with one hidden spin unit and the remaining Gaussian hidden units (Fig. 13). First column shows the legend: black for the unconstrained model, red for models with all hidden units constrained, and blue for models with the constraint acting on all but one hidden unit. Subsequent columns show the results for the three datasets considered: MNIST0/1, two-dimensional Ising model ($L = 64$), and the KH protein domain. The discontinuous arrows in the first panel highlight the likelihood costs of partial label erasure (red) and disentanglement (blue).

TABLE I. Decrease of log-likelihoods corresponding to partial erasure of the label with a fully constrained RBM $\Delta\mathcal{L}_{\text{part erasure}}$, and to disentanglement with a partially constrained RBM $\Delta\mathcal{L}_{\text{disent}}$. The changes on log-likelihoods are expressed per data configuration and per pixel for MNIST0/1, per spin for 2D Ising, and per protein site for the KH domain.

Model	Label	$\Delta\mathcal{L}_{\text{part erasure}}$	% of unconstrained log-likelihood	$\Delta\mathcal{L}_{\text{disent}}$	% of unconstrained log-likelihood
MNIST0/1	0 or 1	0.016	30%	0.005	10%
2D Ising ($L = 64$, $\beta = 0.44$)	Sign of magnetization	0.18	40%	$\simeq 0$	$\simeq 0\%$
KH domain	Bacteria or eukaryotic	0.09	6%	0.04	3%

two bounds sandwiches the true likelihood value and ensures that sampling has converged.

The results are shown for the Ising model, MNIST0/1, and PF00013 datasets considered in this work in the top row of Fig. 12. We do not consider CelebA for computational convenience. We first measure the likelihood costs $\Delta\mathcal{L}_{\text{part erasure}}$ [see Eq. (11)] for making labels inaccessible to linear discriminators with the fully constrained architecture (red bars or dots). In all datasets, the labels considered are relevant to the nature of the data, and the costs (per data configuration) induced by the constraints on the weights are significant; see Table I.

The relation between label relevance and the likelihood cost is nicely portrayed in the two-dimensional Ising model dataset. At low β , the data are essentially random, and the magnetization is mostly irrelevant to determining the probability of a configuration. In this regime, erasing label information has little likelihood cost. As the inverse temperature increases, the magnetization becomes more relevant, and it becomes necessary for the model to account for it to achieve good likelihood. In consequence, partially erasing the magnetization in this regime results in a large likelihood loss.

The top row of Fig. 12 furthermore shows the values of the log-likelihoods after releasing one hidden unit (blue bars and dots). The log-likelihood loss with respect to the unconstrained RBM $\Delta\mathcal{L}_{\text{disent}}$ in Eq. (12) is guaranteed to be non-negative. In practice, for the MNIST0/1 and Ising model datasets, and to a lesser extent for the KH domain, we estimate this cost to be small; see Table I. These results are consistent with the ability of the released RBM to fit and generate high-quality data in the three cases, as shown in previous sections.

B. Analytical estimates

We can gain some analytical insights into the origin of the costs of partial erasure and disentanglement as follows. To make our RBM models mathematically tractable, we now assume that the visible and hidden units of the RBM are all real valued and Gaussianly distributed, with the exception of a single spinlike hidden unit, $h^* = h_1 = \pm 1$ (intended to be eventually released to help concentrate the

label-related information). This RBM model defines a bimodal Gaussian mixture distribution, with two modes associated with the label classes $u = \pm 1$; see Figs. 13(a) and 13(b).

The energy function under this Gaussian-spin (GS) RBM model writes

$$E_{\text{GS}}(\mathbf{v}, \mathbf{h}) = \sum_i \frac{v_i^2}{2\sigma_i^2} - \sum_i g_i v_i + \sum_{\mu \geq 2} \frac{h_\mu^2}{2} - \sum_i \sum_{\mu \geq 2} w_{i\mu} v_i h_\mu - \sum_i w_i^* v_i h_1, \quad (31)$$

where the σ_i 's parametrize the standard deviations of the visible units, and the visible units are connected to the Gaussian hidden units through the weights $w_{i\mu}$ and to the spin hidden unit through w_i^* .

We first train the RBM in the absence of any constraint on the weights. The data are characterized by their empirical correlation matrix \mathbf{C} and the vector $\mathbf{q}^{(1)}$ separating the centers of mass of the classes; see Fig. 2(c). Maximizing the likelihood of the data gives several conditions over the weight vectors that we list below.

- (1) The scaled weights $w_{i\mu}\sigma_i$ for $\mu \geq 2$ are eigenvectors of the matrix $\tilde{\mathbf{C}} = \mathbf{D}(\mathbf{C} - \mathbf{q}^{(1)}(\mathbf{q}^{(1)})^\top)\mathbf{D}$, with corresponding eigenvalues $\lambda_\mu = 1/(1 - \sum_i w_{i\mu}^2 \sigma_i^2)$; here, \mathbf{D} is the diagonal matrix with entries $1/\sigma_i^2$. In practice, the top $M - 1$ eigenvalues of $\tilde{\mathbf{C}}$ (larger than unity) have to be selected to maximize the likelihood.

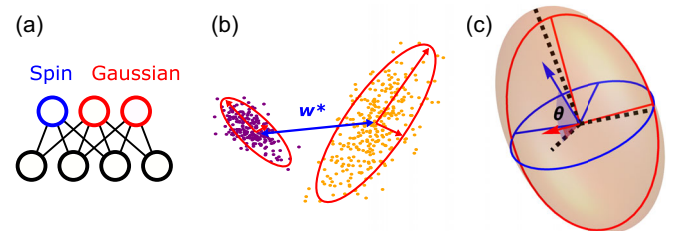


FIG. 13. Gaussian-spin RBM. (a) The Gaussian-spin RBM has one spinlike hidden unit $h^* = h_1 = \pm 1$, whereas all other hidden units are Gaussian. (b) The spin hidden unit (blue) separates the two labeled classes. Gaussian hidden units (red) model intraclass variability. (c) Illustration of Poincaré theorem.

- (2) The weights \mathbf{w}^* onto hidden unit $\mu = 1$ are given by $\mathbf{\Sigma}^{-1}\mathbf{q}^{(1)}$, where $\mathbf{\Sigma} = (\mathbf{D} - \mathbf{W}\mathbf{W}^\top)^{-1}$ denotes the conditional covariance matrix predicted by the model within each class, and \mathbf{W} is the matrix of weight vectors $w_{i\mu}$ with $\mu \geq 2$.
- (3) The biases on the visible units are such that the model fits the independent site frequencies: $\mathbf{g} = \mathbf{\Sigma}^{-1}(\langle \mathbf{v} \rangle_{\mathcal{D}} - \mathbf{q}^{(1)})$.

Details about the derivation can be found in Supplemental Material [26] Appendix D. The log-likelihood reads

$$L_{\text{GS}} = \frac{1}{2} \sum_{\mu} (\lambda_{\mu} - 1 - \log \lambda_{\mu}) - \log \cosh(\mathbf{g} \cdot \mathbf{q}^{(1)}), \quad (32)$$

where the λ_{μ} 's are the selected eigenvalues of $\tilde{\mathbf{C}}$, and we ignore irrelevant additive terms.

We next consider maximum likelihood training of a RBM in the presence of orthogonality constraints acting on the Gaussian weights, while w_i^* is unconstrained; see Eq. (6). Let us define the projection operator onto the subspace orthogonal to $\mathbf{q}^{(1)}$,

$$\mathbf{P} = \mathbb{I} - \frac{\mathbf{q}^{(1)}(\mathbf{q}^{(1)})^\top}{|\mathbf{q}^{(1)}|^2}. \quad (33)$$

It is easy to realize that conditions (6) are equivalent to $\mathbf{P}\mathbf{W} = \mathbf{W}$. Consequently, the discussion of the unconstrained learning case above applies to the constrained case provided the correlation matrix $\tilde{\mathbf{C}}$ is replaced with the projected matrix $\tilde{\mathbf{C}}^\perp = \mathbf{P}\tilde{\mathbf{C}}\mathbf{P}$.

The eigenvalues of the projected matrix $\tilde{\mathbf{C}}^\perp$ have a precise ordering relationship to the eigenvalues of the original matrix $\tilde{\mathbf{C}}$ known as Poincaré separation theorem (see Theorem 11.11 of Ref. [22]). Denoting by $\lambda_1, \dots, \lambda_N$ the eigenvalues of the original matrix, and $\lambda_1^\perp, \dots, \lambda_N^\perp$ the eigenvalues of the projected matrix, both ranked in decreasing order, we have

$$\lambda_1 \geq \lambda_1^\perp \geq \lambda_2 \geq \lambda_2^\perp \geq \dots \geq \lambda_N \geq \lambda_N^\perp = 0, \quad (34)$$

where $\lambda_N^\perp = 0$ is due to the forbidden direction $\mathbf{q}^{(1)}$, which results in a drop of the rank of the matrix. Moreover, the gaps $\lambda_i - \lambda_i^\perp$ are connected to the angle between the forbidden direction $\mathbf{q}^{(1)}$ and the eigenvectors of the original correlation matrix. Figure 13(c) shows a low-dimensional example, in which a three-dimensional ellipsoid symbolizing $\tilde{\mathbf{C}}$ is projected to the space orthogonal to one of the vectors shown. We consider two vectors with different angles to the ellipsoid principal axis, which define the projected ellipse $\tilde{\mathbf{C}}^\perp$.

The likelihood of the released Gaussian-spin RBM is given by the same formula as for the unconstrained model [see Eq. (32)] upon replacement $\lambda_{\mu} \rightarrow \lambda_{\mu}^\perp$. As the function

is monotonic in the eigenvalues (when they are larger than unity), Poincaré separation theorem in Eq. (34) guarantees that the likelihood decreases when imposing the constraints on the weights.

Lastly, when the orthogonality constraint (6) acts on all weights, the model is blind to the separation of the classes. We obtain the likelihood of the constrained RBM by simply replacing $\mathbf{q}^{(1)}$ in the above calculation with the zero vector, and consequently, $w_i^* = 0$ also.

The bottom row of Fig. 12 shows the log-likelihood estimates produced by this approximate calculation in the unconstrained, constrained, and released cases. While the absolute values of the log-likelihoods cannot be directly compared to the binary RBM settings, we see that the relative changes from unconstrained to constrained associated with the partial erasure cost, and from constrained to released defining the disentanglement cost fairly match their counterparts computed by annealed importance sampling on binary RBMs.

X. DISCUSSION

In this work, we propose computationally efficient methods to train RBMs with disentangled representations. In turn, these representations can be used to generate samples with desired properties, e.g., with one attribute changed while the other features remain unaffected. This goal has been pursued in the literature [7–9,11] with deep neural networks, predominantly with variational autoencoders (VAEs) [3,54] and adversarial networks [4,7,11]. Despite the broad success of adversarial learning and its importance in practical applications [7], the aforementioned methods suffer from several drawbacks. Deep neural networks are difficult to interpret and require large amounts of data to train. Variational autoencoders [3] enforce a continuous mapping of the data to a Gaussian distribution, which is not always suitable, for instance, if the data consist of separated peaks [55]. Last of all, adversarial training suffers from instabilities that are not fully understood yet, making training difficult to implement in practice.

Our approach exploits the simplicity of the RBM architecture. Despite the limited number of layers, the flexibility in the potentials on hidden units allows RBMs to express complex representation distributions, contrary to VAEs that require deeper architectures to map the data distribution onto Gaussian latent variables. We derive explicit constraints to be applied to the RBM weights during learning to favor disentangled representations. These constraints enforce that the data representations corresponding to different label classes are approximately indistinguishable. More precisely, we impose linear and quadratic constraints on the RBM weights that (partially) decorrelate the class label from the hidden-unit activities. As in an adversarial framework, imposing these constraints on a subset of hidden units allows us to manipulate the

samples generated from the model by controlling the state of the remaining hidden units.

The resulting training algorithm is easily implementable and fast, being based on two steps. First, we estimate the required constraints from labeled data. Crucially, this is the only step that requires labels. Second, we train the RBM with standard learning procedures [56], making sure that, after each gradient update, the weights are projected into the subspace satisfying the constraints. The resulting procedure has a similar computational cost to standard RBM training. It is therefore robust, not suffering from instability due to the maximization-minimization of the cost function appearing in adversarial learning schemes. We again stress that our approach combines the unsupervised nature of the RBM with constraints that are derived from labeled data. Therefore, our model can be said to be semisupervised. We show how this synergy results in a model able to work in a regime with a limited amount of labeled data. This result is important as, in many cases, labeled data are much more expensive to obtain than unlabeled data: Data have to be annotated by humans (for instance, in the PF00013 dataset of the KH domain sequences, taxonomy labels are available for less than 10% of the sequences), or costly experiments have to be done to get the label (this is the case for most biological data, which often require complex biophysical or biochemical characterizations).

We demonstrate the effectiveness of this approach on four datasets from diverse domains: the CelebA dataset of face images [23], the Ising model from statistical physics, the MNIST collection of handwritten digit images [24], and protein sequences of the KH domain family [25].

CelebA [23] and MNIST [24] are popular benchmark datasets in machine learning. In MNIST, the labels are straightforwardly associated with the digit identities. On this dataset, we show that RBM can be trained to associate one or few controlling hidden units with each digit class, which can be manipulated to sample and transition between classes. In CelebA, the labels correspond to subtle attributes of face images, like facial expressions (smiling or not smiling), or adornments (presence of eyeglasses). Even for this complex dataset, RBMs can sample good-looking images and are able to concentrate these attributes over few hidden units.

The two-dimensional Ising model is a very well-studied system in statistical physics, with a precisely characterized phase transition controlled by the temperature. A standard RBM is able to reproduce the behaviors of observables, such as the magnetization, heat capacity, susceptibility, and correlation length. We then impose a linear constraint on the weights [see Eq. (6)], decorrelating the latent representation from the magnetization sign and forcing the RBM to hallucinate a new system with interesting physical properties. Remarkably, the constrained RBM generates configurations with zero net magnetization, while

preserving the structure of correlations between spins, as evident from second-order observables, such as the heat capacity and correlation length. Through a heuristic argument, we propose a Hamiltonian to describe the physical properties of this system, containing a nonanalytic penalty term for the global magnetization reminiscent of nonanalytic Landau potentials recently proposed to describe nonequilibrium steady states of the Ising magnet [57–59]. Releasing a single hidden unit then restores the ability of the model to generate magnetized configurations, reproducing all statistics of the original Ising model.

Our last application is in protein design based on model learning from sequence data, a field which has grown in importance in bioengineering since the recent impressive developments of sequencing technologies [60]. RBMs trained on the K -homology domain family under linear constraints decorrelating a subset of hidden inputs from the taxonomy of sequences, efficiently concentrate taxonomic information in a control hidden unit. Conditional sampling reproduces the fine statistical differences of the eukaryotic and bacterial subfamilies. The transition between the two classes takes place in a shorter time than the overall decorrelation time, suggesting that sequences might be able to change class while maintaining a memory of other, class-independent attributes.

Concentrating information about important features of the data into one or few hidden units of the RBM could *a priori* be detrimental to the ability of the model to fit the data for two reasons. First, introducing constraints on the weights is expected to impact (decrease) the log-likelihood of the data generated by the RBM. We estimate the losses in log-likelihood due to partial erasure and disentanglement for several datasets. The cost of partial erasure is related to the relevance of the label, as clearly illustrated in the dependence on temperature in the Ising model data. Remarkably, we find that disentanglement is achieved with a small relative likelihood loss, evidencing the robustness of the approach. Furthermore, when the data can be approximated as a mixture of two Gaussian distributions, we show how the log-likelihood losses could be analytically calculated and establish a connection between the likelihood costs for erasure or disentanglement and the Poincaré separation theorem.

Second, the few (often, single) released hidden units encode label-associated features in a prototypelike way. In the case of linear constraints, released weights are aligned with the $\mathbf{q}^{(1)}$ vector, equal to the relative difference between the centers of mass of the two label classes; see Fig. 5(a) for an illustration on CelebA. It is however widely believed that prototypelike representations are poorer than compositional ones, in which multiple features associated with many hidden units can be combinatorially combined to create high-quality and diverse data [61]. From this point of view, forcing some hidden units to generate prototypes could appear counterproductive. It is nevertheless a very

effective way to drive class switching; see, for instance, Fig. 6. In addition, all the important features defining the data distribution are learned by the vast number of other (constrained) hidden units, which, in turn, can be combined together to collectively participate in the data generation process. We also emphasize that, while a few hidden units capture enough label-associated features to manipulate and drive the label values, this does not mean that they concentrate all the information about the label. As clearly shown in Fig. 7(f) for Ising and Fig. 8(b) for MNIST0/1, there remains substantial information about the label in the constrained hidden units accessible to deep decoders. Hence, label-associated features are residually encoded in a combinatorial way by the RBM.

While disentangling and manipulating representations through our “partially constrained” RBM approach offers clear advantages in terms of usability and interpretability, the other architecture we consider in this work, the so-called “fully constrained” RBM may also be of interest in practical applications. Informally speaking, fully constrained RBMs are appropriate to model the features in the data orthogonal to the ones associated with the label under consideration. We show that fully constrained RBMs remain generative in two examples (CelebA and PF00013), where samples resemble data configurations with ambiguous class identity. In the MNIST0/1 and Ising model examples, however, the fully constrained RBM generates samples markedly different from the data (zero magnetization in the Ising case, and blurry mixtures of 0s and 1s for MNIST). We attribute this to the fact that in these latter cases, the datasets corresponding to the two values of the label are widely separated. However, as we show in the Ising case, information is preserved in higher-order moments of the samples (e.g., heat capacity). Another example is shown in Supplemental Material [26] Fig. S6, where a fully constrained RBM trained on zero MNIST digits in black or white backgrounds generates zeros encoded in the correlations between neighboring pixels. As a potential future direction for fully constrained RBM, our results on the KH domain open the way to the reconstruction of ancestral (backward in evolutionary time) proteins, which were possibly more functionally promiscuous than their current counterparts. It would be very interesting to apply our approach to reconstruct putative ancient proteins, e.g., where details about binding specificity are erased while the other functionalities (stability, activity, etc.) are maintained.

In summary, our work proposes a flexible semisupervised framework for learning disentangled representations, easily implementable and amenable to approximate analytical calculations. We hope our approach will make controlled generation of data and feature discovery easier in future applications. Last of all, besides the applications to RBMs we present here, it would be interesting to transfer our constraint-based framework to other architectures, as

the principle of imposing constraints on the weights in the course of learning is quite general.

The codes needed to reproduce the results reported in this work are available on [62].

ACKNOWLEDGMENTS

J. F.-d.-C.-D., S. C., and R. M. are supported by Grants No. ANR-19 Decrypted CE30-0021-01, and No. ANR-21 Locomat CE16-0037.

-
- [1] Y. Bengio, *Deep Learning of Representations for Unsupervised and Transfer Learning*, in *Proceedings of the ICML Workshop on Unsupervised and Transfer Learning* (PMLR, 2012), pp. 17–36.
 - [2] R. Salakhutdinov and G. Hinton, *Deep Boltzmann Machines*, in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics* (PMLR, 2009), pp. 448–455.
 - [3] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
 - [4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Networks*, [arXiv:1406.2661](https://arxiv.org/abs/1406.2661).
 - [5] W. J. Johnston, S. E. Palmer, and D. J. Freedman, *Nonlinear Mixed Selectivity Supports Reliable Neural Computation*, *PLoS Comput. Biol.* **16**, e1007544 (2020).
 - [6] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, *Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations*, in *Proceedings of the 36th International Conference on Machine Learning* (PMLR, 2019), pp. 4114–4124.
 - [7] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, *Fader Networks: Manipulating Images by Sliding Attributes*, [arXiv:1706.00409](https://arxiv.org/abs/1706.00409).
 - [8] H. Kim and A. Mnih, *Disentangling by Factorising*, in *Proceedings of the 35th International Conference on Machine Learning* (PMLR, 2018), pp. 2649–2658.
 - [9] Q. Hu, A. Szabó, T. Portenier, P. Favaro, and M. Zwicker, *Disentangling Factors of Variation by Mixing Them*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, New York, 2018), pp. 3399–3407.
 - [10] B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy, and J.-W. Meent, *Structured Disentangled Representations*, in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics* (PMLR, 2019), pp. 2525–2534.
 - [11] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, *ATTGAN: Facial Attribute Editing by Only Changing What You Want*, *IEEE Trans. Image Process.* **28**, 5464 (2019).
 - [12] Y. Shen, J. Gu, X. Tang, and B. Zhou, *Interpreting the Latent Space of GANS for Semantic Face Editing*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, New York, 2020), pp. 9243–9252.

- [13] J. Zaidi, J. Boilard, G. Gagnon, and M.-A. Carboneau, *Measuring Disentanglement: A Review of Metrics*, [arXiv:2012.09276](#).
- [14] C. Feutry, P. Piantanida, Y. Bengio, and P. Duhamel, *Learning Anonymized Representations with Adversarial Neural Networks*, [arXiv:1802.09386](#).
- [15] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, *Learning Fair Representations*, in *Proceedings of the 30th International Conference on Machine Learning* (PMLR, 2013), pp. 325–333.
- [16] M. Arjovsky and L. Bottou, *Towards Principled Methods for Training Generative Adversarial Networks*, [arXiv:1701.04862](#).
- [17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, [arXiv:1301.3781](#).
- [18] G. E. Hinton, *A Practical Guide to Training Restricted Boltzmann Machines*, in *Neural Networks: Tricks of the Trade* (Springer, New York, 2012), pp. 599–619.
- [19] J. Tubiana, S. Cocco, and R. Monasson, *Learning Protein Constitutive Motifs from Sequence Data*, *eLife* **8**, e39397 (2019).
- [20] B. Bravi, J. Tubiana, S. Cocco, R. Monasson, T. Mora, and A. M. Walczak, *RBM-MHC: A Semi-Supervised Machine-Learning Method for Sample-Specific Prediction of Antigen Presentation by HLA-I Alleles*, *Cell Syst.* **12**, 195 (2021).
- [21] R. Salakhutdinov, A. Mnih, and G. Hinton, *Restricted Boltzmann Machines for Collaborative Filtering*, in *Proceedings of the 24th International Conference on Machine Learning* (PMLR, 2007), pp. 791–798.
- [22] K. M. Abadir and J. R. Magnus, *Matrix Algebra* (Cambridge University Press, Cambridge, England, 2005), Vol. 1.
- [23] Z. Liu, P. Luo, X. Wang, and X. Tang, *Deep Learning Face Attributes in the Wild*, in *Proceedings of International Conference on Computer Vision* (2015), https://openaccess.thecvf.com/content_iccv_2015/html/Liu_Deep_Learning_Face_ICCV_2015_paper.html.
- [24] L. Deng, *The MNIST Database of Handwritten Digit Images for Machine Learning Research*, *IEEE Signal Process. Mag.* **29**, 141 (2012).
- [25] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart *et al.*, *The Pfam Protein Families Database in 2019*, *Nucleic Acids Res.* **47**, D427 (2019).
- [26] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevX.13.021003> for implementation details, supplementary text and figures.
- [27] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics* (Elsevier, New York, 2016).
- [28] D. Yevick and R. Melko, *The Accuracy of Restricted Boltzmann Machine Models of Ising Systems*, *Comput. Phys. Commun.* **258**, 107518 (2021).
- [29] M. Harsh, J. Tubiana, S. Cocco, and R. Monasson, *‘Place-Cell’ Emergence and Learning of Invariant Data with Restricted Boltzmann Machines: Breaking and Dynamical Restoration of Continuous Symmetries in the Weight Space*, *J. Phys. A* **53**, 174002 (2020).
- [30] T. M. Cover, *Elements of Information Theory* (John Wiley & Sons, New York, 1999).
- [31] A. Engel and C. Van den Broeck, *Statistical Mechanics of Learning* (Cambridge University Press, Cambridge, England, 2001).
- [32] N. Brenner, W. Bialek, and R. d. R. Van Steveninck, *Adaptive Rescaling Maximizes Information Transmission*, *Neuron* **26**, 695 (2000).
- [33] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press, Cambridge, MA, 2018).
- [34] V. A. Marchenko and L. A. Pastur, *Distribution of Eigenvalues for Some Sets of Random Matrices*, *Mat. Sb.* **114**, 507 (1967).
- [35] A. Decelle, C. Furtlehner, and B. Seoane, *Equilibrium and Non-Equilibrium Regimes in the Learning of Restricted Boltzmann Machines*, *Adv. Neural Inf. Process. Syst.* **34**, 5345 (2021).
- [36] L. Onsager, *Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition*, *Phys. Rev.* **65**, 117 (1944).
- [37] M. E. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, Oxford, 1999).
- [38] N. Yoshioka, Y. Akagi, and H. Katsura, *Transforming Generalized Ising Models into Boltzmann Machines*, *Phys. Rev. E* **99**, 032113 (2019).
- [39] G. Cossu, L. Del Debbio, T. Giani, A. Khamseh, and M. Wilson, *Machine Learning Determination of Dynamical Parameters: The Ising Model Case*, *Phys. Rev. B* **100**, 064304 (2019).
- [40] W. Selke, *Critical Binder Cumulant of Two-Dimensional Ising Models*, *Eur. Phys. J. B* **51**, 223 (2006).
- [41] K. Shimagaki and M. Weigt, *Selection of Sequence Motifs and Generative Hopfield-Potts Models for Protein Families*, *Phys. Rev. E* **100**, 032128 (2019).
- [42] N. V. Grishin, *KH Domain: One Motif, Two Folds*, *Nucleic Acids Res.* **29**, 638 (2001).
- [43] B. M. Lunde, C. Moore, and G. Varani, *RNA-Binding Proteins: Modular Design for Efficient Function*, *Nat. Rev. Mol. Cell Biol.* **8**, 479 (2007).
- [44] R. Valverde, L. Edwards, and L. Regan, *Structure and Function of KH Domains*, *FEBS J.* **275**, 2712 (2008).
- [45] G. Musco, G. Stier, C. Joseph, M. A. C. Morelli, M. Nilges, T. J. Gibson, and A. Pastore, *Three-Dimensional Structure and Stability of the KH Domain: Molecular Insights into the Fragile X Syndrome*, *Cell* **85**, 237 (1996).
- [46] W. T. O’Donnell and S. T. Warren, *A Decade of Molecular Studies of Fragile X Syndrome*, *Annu. Rev. Neurosci.* **25**, 315 (2002).
- [47] D. Sehnal, S. Bittrich, M. Deshpande, R. Svobodová, K. Berka, V. Bazgier, S. Velankar, S. K. Burley, J. Koča, and A. S. Rose, *Mol* Viewer: Modern Web App for 3D Visualization and Analysis of Large Biomolecular Structures*, *Nucleic Acids Res.* **49**, W431 (2021).
- [48] T. U. Consortium, *UniProt: The Universal Protein Knowledgebase in 2021*, *Nucleic Acids Res.* **49**, D480 (2020).
- [49] S. Cocco, C. Feinauer, M. Figliuzzi, R. Monasson, and M. Weigt, *Inverse Statistical Physics of Protein Sequences: A Key Issues Review*, *Rep. Prog. Phys.* **81**, 032601 (2018).
- [50] 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*, *Proc. Natl. Acad. Sci. U.S.A.* **108**, E1293 (2011).
- [51] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, and M. Steinegger, *COLABFOLD: Making Protein Folding Accessible to All*, *Nat. Methods* **19**, 679 (2022).
- [52] R. Neal, *Annealed Importance Sampling*, Department of Statistics, University of Toronto Technical Report No. 9805 (revised), 1998.
- [53] Y. Burda, R. Grosse, and R. Salakhutdinov, *Accurate and Conservative Estimates of MRF Log-Likelihood Using Reverse Annealing*, in *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics* (PMLR, 2015), pp. 102–110, <http://proceedings.mlr.press/v38/burda15.html>.
- [54] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, *BETA VAE: Learning Basic Visual Concepts with a Constrained Variational Framework*, in *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- [55] S. Goldt, B. Loureiro, G. Reeves, F. Krzakala, M. Mézard, and L. Zdeborová, *The Gaussian Equivalence of Generative Models for Learning with Shallow Neural Networks*, in *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference* (PMLR, 2022), pp. 426–471.
- [56] T. Tieleman, *Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient*, in *Proceedings of the 25th International Conference on Machine Learning* (PMLR, 2008), pp. 1064–1071.
- [57] D. Belitz, T. R. Kirkpatrick, and T. Vojta, *How Generic Scale Invariance Influences Quantum and Classical Phase Transitions*, *Rev. Mod. Phys.* **77**, 579 (2005).
- [58] C. Aron and M. Kulkarni, *Nonanalytic Nonequilibrium Field Theory: Stochastic Reheating of the Ising Model*, *Phys. Rev. Res.* **2**, 043390 (2020).
- [59] C. Aron and C. Chamon, *Landau Theory for Non-Equilibrium Steady States*, *SciPost Phys.* **8**, 074 (2020).
- [60] H. T. Rube, C. Rastogi, S. Feng, J. F. Kribelbauer, A. Li, B. Becerra, L. A. Melo, B. V. Do, X. Li, H. H. Adam *et al.*, *Prediction of Protein-Ligand Binding Affinity from Sequencing Data with Interpretable Machine Learning*, *Nat. Biotechnol.* **40**, 1520 (2022).
- [61] J. Tubiana and R. Monasson, *Emergence of Compositional Representations in Restricted Boltzmann Machines*, *Phys. Rev. Lett.* **118**, 138301 (2017).
- [62] <https://github.com/cossio/AdvRBMs.jl>