PAPER • OPEN ACCESS

Unlearning regularization for Boltzmann machines

To cite this article: Enrico Ventura *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 025078

View the article online for updates and enhancements.

PAPER

Unlearning regularization for Boltzmann machines

Enrico Ventura[1,2], Simona Cocco[2], Rémi Monasson[2] and Francesco Zamponi[1,2,*] (iD)

1 Dipartimento di Fisica, Sapienza Universitá di Roma, P.le A. Moro 2, 00185 Roma, Italy
2 Laboratoire de Physique de l'Ecole Normale Supérieure, ENS, Université PSL, F-75005 Paris, France
* Author to whom any correspondence should be addressed.

E-mail: francesco.zamponi@uniroma1.it

Abstract

Boltzmann machines (BMs) are graphical models with interconnected binary units, employed for the unsupervised modeling of data distributions. When trained on real data, BMs show the tendency to behave like critical systems, displaying a high susceptibility of the model under a small rescaling of the inferred parameters. This behavior is not convenient for the purpose of generating data, because it slows down the sampling process, and induces the model to overfit the training-data. In this study, we introduce a regularization method for BMs to improve the robustness of the model under rescaling of the parameters. The new technique shares formal similarities with the unlearning algorithm, an iterative procedure used to improve memory associativity in Hopfield-like neural networks. We test our unlearning regularization on synthetic data generated by two simple models, the Curie–Weiss ferromagnetic model and the Sherrington–Kirkpatrick spin glass model. We show that it outperforms L_p-norm schemes and discuss the role of parameter initialization. Eventually, the method is applied to learn the activity of real neuronal cells, confirming its efficacy at shifting the inferred model away from criticality and coming out as a powerful candidate for actual scientific implementations.

1. Introduction

Boltzmann machines (BMs) [1–3] are a class of graphical models, capable of modelling data distributions through the learning of effective pairwise interactions between variables. Once properly trained e.g. through gradient ascent of the likelihood, BMs can be used to generate new data configurations, which hopefully are indistinguishable from the ones in the dataset [4]. Good generative performances are, in practice, hindered by different limitations of the inferential problem.

While in theory BMs (with a sufficient number of hidden units) are universal approximators, in practical applications one is limited by the incompatibility between the true unknown distribution of the training data and the energy-based distribution that is employed to fit them. As a consequence, the training process cannot converge to the best possible generative performance, and only selects a good approximate model among many possible similarly good choices [5]. Hence, regularization can be used to push the training process to yield specific desired properties, useful for particular applications of the data analysis. Standard schemes for regularization include L_2- and L_1-norm penalties imposed on the pairwise interactions. However, these regularization schemes have a tendency to decrease the values of the interactions, and introduce systematic biases in the model.

Another limitation, which is intrinsic in inferential problems, is the limited availability of training data, possibly leading to the overfitting phenomenon. Informally speaking, overfitting is an over-specialization of the model, which is too focused on the details of the specific training set, rather than capturing the broader statistical structure highlighted by the data. Overfitting is particularly concerning in the so-called high dimensional setting, in which the size of the model (the dimension of the data configuration) is comparable to the amount of available data. Regularization is also supposed to help in avoiding overfitting.

Due to regularization, in a statistical physics language, the inferred interactions become smaller, and the inferred model is implicitly at higher temperature. Correcting this bias requires introducing a temperature parameter smaller than unity when generating new data. This procedure was used in [6] to generate new viable protein sequences with BMs inferred from evolutionary sequence data. However, this approach is purely empirical and somewhat uncontrolled. It is known from the statistical physics literature that rescaling the energy function by a temperature factor, even close to one, can drastically alter the distribution properties if the system is near a phase transition point [7, 8]. Hence, inferring models that are robust—i.e. generate similar data—when the energy function is rescaled might be extremely advantageous.

The goal of this study, motivated by the practical issues described above, is to propose a new regularization technique for BM learning that explicitly aims at increasing the robustness of the model under rescaling of its parameters, i.e. the coupling matrix $\boldsymbol{J} = \{J_{ij}\}$ and the local fields $\boldsymbol{h} = \{h_i\}$. This new tool, that we named unlearning regularization, appears to be more effective than other techniques, approaching a higher robustness performance of the neural network. The performance of the unlearning regularization is evaluated from artificial data generated by both a Curie–Weiss (CW) and a Sherrington–Kirkpatrick (SK) models. We initially consider a small number of variables: in this way the original parameters are known, the fixed points of the learning equations are reached precisely and the useful quantities can be computed exactly. In addition, higher-dimensional and more realistic cases are also considered to validate the results.

A particular limit of this type of regularization coincides with a thermal version of the traditional Hebbian Unlearning algorithm (HU) [9–12]. We thus specifically consider this limit and conclude that HU is able to infer the original data distribution with a very good accuracy. The performance shows an optimum in time that scales with the control parameters, as it has been observed when the algorithm is implemented in associative memory tasks [13]. We conclude that HU can be interpreted as a two-steps BM learning procedure.

The structure of the article is the following. We first introduce generative modeling and BMs in section 2. Some extensively employed regularization techniques (i.e. the L_p methods) are then described in section 2.2. Section 3 briefly describes the HU algorithm and its traditional use in associative memory models. The new unlearning regularization is then defined in section 3.2. Its performance is analyzed for two different data sources in section 4: a CW model (section 4.1) and a SK model (section 4.2–4.5). In section 4.3–4.5 we compare the new regularization method to the standard L_p regularization scheme, showing that it substantially improves the robustness under rescaling of the parameters while preserving a high similarity to the original model. The analysis is performed both at convergence of the regularization algorithm and in the condition of highest similarity with the ground-truth model, obtained via early-stopping in the training, either in the small and large N cases. The study of a particular limit of the regularization technique leading to HU follows in section 5, which provides some insight on its behavior as a generic inference tool. Eventually, the unlearning method is tested on real biological data in section 6, confirming in practice the validity of our new regularization prescription.

2. Generative modeling

Generative models are a class of specific neural networks that are able to learn the probability distribution of a data-set and generate brand new data that exhibit maximum coherence with the same statistics [3, 4].

Consider a collection of N variables in a vector $\vec{S} = (S_1, \ldots, S_N)$. Data are M realizations of such a vector grouped into a set $\{\vec{S}^\mu\}_{\mu=1}^M$ and we assume that they are sampled independently from a joint distribution $P_{\text{true}}(\vec{S})$. Given M data, we have access to the frequency of occurrence of a certain variable, i.e. the empirical distribution

$$P_{\text{data}}\left(\vec{S}\right) = \frac{1}{M} \sum_{\mu=1}^M \prod_{i=1}^N \delta_{S_i^\mu, S_i}. \tag{1}$$

Generative modeling looks for a model distribution $P_{\text{mod}}(\vec{S}|\hat{\theta})$, where the parameters $\hat{\theta}$ are inferred from the training data such that P_{mod} is the closest possible to P_{data}. However, P_{data} might still differ from the ground-truth distribution P_{true}. As a remedy, it is useful to reduce the number of degrees of freedom of the problem by designing the model taking into account some guess we might have about P_{true}. For instance one might choose a *graphical model*, where variables are nodes of a graph [3] with interactions to be inferred. Alternatively, one might add a prior distribution for the variables, such as a mixture of Gaussians of unknown means and unit variances [14]. The generative approach is largely implemented across several disciplines such as computational neuroscience [15, 16], bio-informatics [17], animal behavior [18], physical simulations [19], image and text synthesis [20–22].

2.1. BMs

We now describe a specific graphical model of generative neural networks, which will be particularly relevant to the present work. It is inspired by equilibrium statistical mechanics and it is called Boltzmann machine (BM) [1–3, 23].

Consider a fully connected network of N binary Ising variables $\vec{S} \in \{-1, +1\}^N$ with the following energy function

$$E\left[\vec{S}|\boldsymbol{J},\boldsymbol{h}\right] = -\sum_{i,j>i}^{1,N} S_i J_{ij} S_j - \sum_{i=1}^{N} h_i S_i, \tag{2}$$

where J_{ij} are symmetric couplings and h_i are the fields acting on each neuron site. Even if several types of architectures are collected under the title of BMs, we will stick to the simplest model, the one endowed exclusively with visible input neurons that interact reciprocally and are subjected to external fields. In fact, our goal will be analyzing the evolution of the interactions between real neurons, which are not necessarily present in other types of machines.

We know, from statistical mechanics, that such a system at equilibrium at a temperature β^{-1} obeys the following *Gibbs–Boltzmann* joint probability distribution

$$P_{\text{mod}}\left(\vec{S}|\boldsymbol{J},\boldsymbol{h},\beta\right) = \frac{1}{Z_\beta}\exp\left(-\beta E\left[\vec{S}|\boldsymbol{J},\boldsymbol{h}\right]\right), \tag{3}$$

with

$$Z_\beta = \sum_{\vec{S}}\exp\left(-\beta E\left[\vec{S}|\boldsymbol{J},\boldsymbol{h}\right]\right). \tag{4}$$

Training a BM means finding the parameters $\boldsymbol{J}$ and $\boldsymbol{h}$ that minimize the distance between P_{data} given by equation (1) and P_{mod} given by equation (3). Note that as a consequence, β can be set to one in the training without loss of generality. In practice, training can be achieved by minimizing the Kullback–Leibler divergence between P_{data} and P_{mod}, i.e.

$$\mathcal{L}(\boldsymbol{J},\boldsymbol{h}) = \sum_{\vec{S}} P_{\text{data}}\left(\vec{S}\right)\log\left(\frac{P_{\text{data}}\left(\vec{S}\right)}{P_{\text{mod}}\left(\vec{S}|\boldsymbol{J},\boldsymbol{h}\right)}\right), \tag{5}$$

which is equivalent to maximizing the cross-entropy of P_{mod} with respect to the empirical distribution P_{data}. Differentiating equation (5) with respect to J_{ij} and h_i, we obtain the gradient of the loss, i.e.

$$\nabla_{ij}\mathcal{L} = \langle S_i S_j \rangle_{\text{mod}} - \langle S_i S_j \rangle_{\text{data}}, \tag{6}$$

$$\nabla_i\mathcal{L} = \langle S_i \rangle_{\text{mod}} - \langle S_i \rangle_{\text{data}}, \tag{7}$$

where $\langle\,\cdot\,\rangle_{\text{data}}$ and $\langle\,\cdot\,\rangle_{\text{mod}}$ are the averages over the respective probability distributions. Therefore, the parameters can be found by iterating the following gradient descent equations
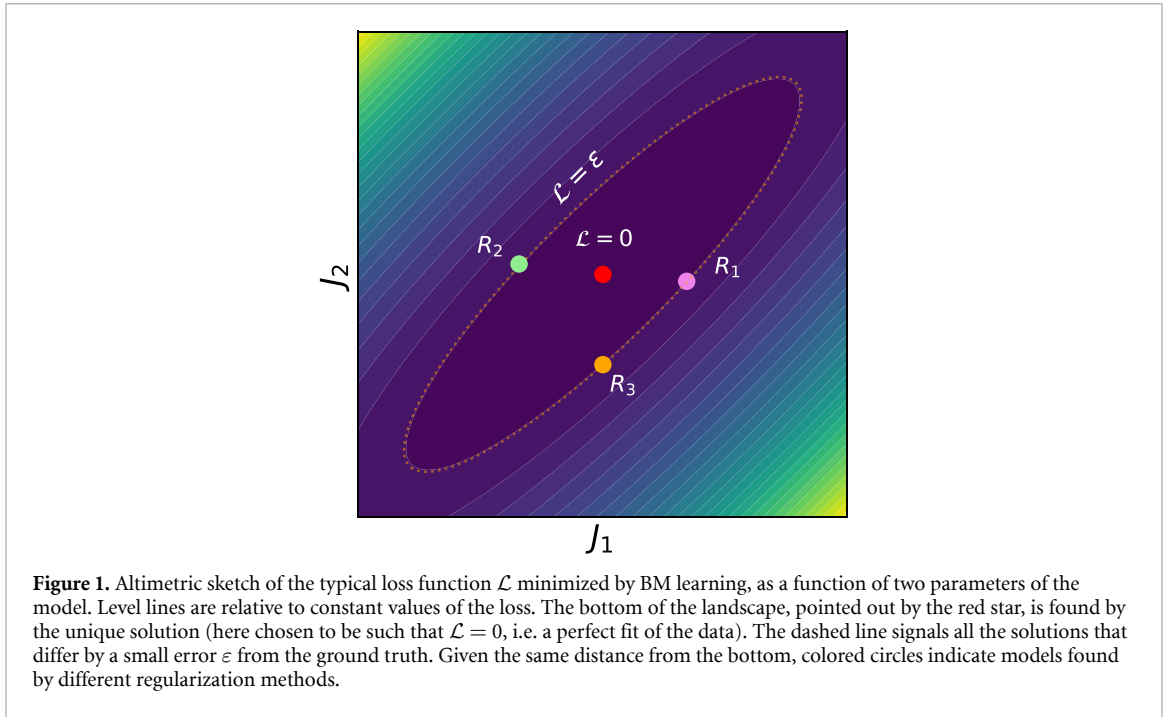
$$\delta J_{ij}^{(t)} = -\lambda\nabla_{ij}\mathcal{L} = \lambda\left(\langle S_i S_j \rangle_{\text{data}} - \langle S_i S_j \rangle_{\text{mod}}\right), \tag{8}$$

$$\delta h_i^{(t)} = -\lambda\nabla_i\mathcal{L} = \lambda\left(\langle S_i \rangle_{\text{data}} - \langle S_i \rangle_{\text{mod}}\right), \tag{9}$$

with λ being a small positive learning rate.

Note that the mean and covariance over the data can be computed upstream, because they only depend on the training data-set; on the other hand, the moments of P_{mod} must be re-sampled at each step of the training process, because their exact calculation would involve a sum over all possible $\vec{S}$. Sampling can be performed by a sufficient number of Monte Carlo chains at equilibrium at $\beta = 1$, which requires an algorithm time that is long enough to ensure ergodicity for each chain. Usually the number of chains should be of the same order of magnitude of M, the number of training data-points, in such a way that the statistical errors on the two averages are comparable.

Once the process converges to the fixed points of equations (8) and (9) we obtain a *moment matching* condition, i.e. the first and the second moments of the two probability distributions coincide. While, in principle, the training of a BM is a convex problem that admits a unique solution [2], in practice there are

Figure 1. Altimetric sketch of the typical loss function $\mathcal{L}$ minimized by BM learning, as a function of two parameters of the model. Level lines are relative to constant values of the loss. The bottom of the landscape, pointed out by the red star, is found by the unique solution (here chosen to be such that $\mathcal{L} = 0$, i.e. a perfect fit of the data). The dashed line signals all the solutions that differ by a small error ε from the ground truth. Given the same distance from the bottom, colored circles indicate models found by different regularization methods.

many flat directions such that some initial conditions can push the parameters closer to their optimal configuration. A common choice is, for instance

$$J_{ij}^{(0)} = \langle S_i S_j \rangle_{\text{data}} - \langle S_i \rangle_{\text{data}} \langle S_j \rangle_{\text{data}}, \tag{10}$$

and

$$h_i^{(0)} = \text{atanh}\left(\langle S_i \rangle_{\text{data}} \right). \tag{11}$$

In order to measure the quality of the training of a BM one can measure whether the moment matching condition is met or not. This can be quantified by the Pearson coefficient between the moments of the P_{mod} and P_{data} distributions at the end of the training. Let us collect the two-points correlation matrices $\langle S_i S_j \rangle$ in a vector $\vec{c}$ where each entry runs over the indices $i, j > i$. Let us group the means $\langle S_i \rangle$ in a similar vector $\vec{\mu}$ where each entry runs over the index i. Then the Pearson coefficients are defined as

$$\rho_J = \frac{\sum_i c_i^{\text{mod}} c_i^{\text{data}}}{\sqrt{\sum_i \left(c_i^{\text{mod}} \right)^2} \sqrt{\sum_i \left(c_i^{\text{data}} \right)^2}}, \tag{12}$$

$$\rho_h = \frac{\sum_i \mu_i^{\text{mod}} \mu_i^{\text{data}}}{\sqrt{\sum_i \left(\mu_i^{\text{mod}} \right)^2} \sqrt{\sum_i \left(\mu_i^{\text{data}} \right)^2}}. \tag{13}$$

2.2. Regularization in BMs

Since the loss in equation (5) is a convex function of the parameters [2], there is a unique Gibbs–Boltzmann distribution that minimizes it. However, in most practical applications the data are limited and as a consequences the moments $\langle \cdot \rangle_{\text{data}}$ are only noisy estimates of the correct ones. Furthermore, the data are not generated by a Gibbs–Boltzmann distribution themselves. Because of this, the loss landscape features many almost flat directions, corresponding to the fact that many different models can fit the data equally well, i.e. reach a loss $\mathcal{L} = \mathcal{L}_{\text{min}} + \varepsilon$ with ε a small KL distance from the ground-truth. This idea is sketched in figure 1. Regularization methods can be used to select specific models in this region of comparable loss, thus providing different inferred systems with different properties, depending on additional requirements imposed on the inferential problem. For instance, some of these models can be sparse or even contain null parameters (e.g. L_0, L_1-norm and information based regularization schemes [24]), others are fully connected (e.g. L_2-norm method). Past literature [7, 8, 25] highlighted that critical models, i.e. inferred systems being highly susceptible to small changes in the parameters, can be attractive for BM learning. In fact, if we consider training as an homogeneous sampling of the models that minimize $\mathcal{L}$, critical models have a large basin of attraction and are thus sampled most often [8]. Criticality can be problematic for data generation: it

implies a long correlation time in the Monte Carlo dynamics, slowing down the sampling process; it makes the model susceptible under rescaling of the parameters, reducing its predictivity in real data applications. Hence, avoiding criticality might help increasing generalization.

To summarize, we define the performance of the model according to two properties: *accuracy*, i.e. the capability of the model to be as similar as possible to the distribution that generated the data; *robustness*, i.e. the tendency of the model not to change its typical configurations when slightly changing (e.g. rescaling by a common temperature factor) the parameters. The goal of this work is to introduce a new type of regularization in order to find a better compromise between these two requirements.

We now discuss some widely used regularization methods for BMs and the way we can benchmark their accuracy and robustness. As mentioned above, L_p regularizations are certainly the most famous regularization methods, also due to their intuitive interpretation. This approach penalizes high values of the parameters by adding a L_p norm of the same variables in the expression of the loss function. As a consequence, the sparsity of the graph is promoted: the redundant parameters are weakened with respect to the relevant ones. The expression of the loss of a BM with L_1 and L_2 regularizations is given by the following equations

$$\mathcal{L}_1 = \mathcal{L}^{\mathrm{BM}}(\boldsymbol{J}, \boldsymbol{h}) + \gamma_1 \sum_{i,j>i} |J_{ij}| + \gamma_1 \sum_i |h_i|, \tag{14}$$

$$\mathcal{L}_2 = \mathcal{L}^{\mathrm{BM}}(\boldsymbol{J}, \boldsymbol{h}) + \frac{\gamma_2}{2} \sum_{i,j>i} J_{ij}^2 + \frac{\gamma_2}{2} \sum_i h_i^2, \tag{15}$$

where $\mathcal{L}^{\mathrm{BM}}$ is defined in equation (5) and γ_1, γ_2 are two regularization rates. This translates into new updating rules for the parameters, i.e.

$$\delta J_{ij}^{(1)} = \delta J_{ij}^{\mathrm{BM}} - \lambda \gamma_1 \mathrm{sign}(J_{ij}), \tag{16}$$

$$\delta h_i^{(1)} = \delta h_i^{\mathrm{BM}} - \lambda \gamma_1 \mathrm{sign}(h_i), \tag{17}$$

$$\delta J_{ij}^{(2)} = \delta J_{ij}^{\mathrm{BM}} - \lambda \gamma_2 J_{ij}, \tag{18}$$

$$\delta h_i^{(2)} = \delta h_i^{\mathrm{BM}} - \lambda \gamma_2 h_i. \tag{19}$$

Accuracy can be quantified by the Kullback–Leibler divergence $D_{\mathrm{KL}}(true|mod)$ between the inferred model and the original one. Though, this quantity is not symmetric under the exchange of the two distributions, which can be inconvenient to define a distance. We can then adopt a symmetric version of the divergence as

$$sD_{\mathrm{KL}}(true, mod) = D_{\mathrm{KL}}(true|mod) + D_{\mathrm{KL}}(mod|true). \tag{20}$$

Even if numerical results do not show a significant difference between these two quantities, we will mainly adopt sD_{KL} in our analysis. Sometimes, the KL divergence is not sufficient to quantify accuracy, so one must compare the model correlations and the data correlations. A detailed analysis of these will be provided in the following.

Regarding *robustness*, it is known that the susceptibility of the system to small variations of temperature can be measured through the *specific heat* [26], defined as

$$C_v(\beta) = -\beta \frac{\partial S_\beta}{\partial \beta} = \beta^2 \left(\langle E^2 \rangle_\beta - \langle E \rangle_\beta^2 \right), \tag{21}$$

where S_β is the entropy of the model at a given inverse temperature β. The fact that inferred neural networks typically display a peak in C_v around the value of β used for training implies the vicinity of a critical point (the finite size of the system impedes C_v to properly diverge and to show a real criticality). Therefore, since all parameters naturally scale with β in equation (3), C_v is a measure of the sensitivity of the model to a small perturbation of the parameters: high values of C_v imply a strong variation of the model entropy under a small variation of the parameters, which suggests strong variations of the inferred statistics as well. Note that other kind of susceptibilities can be defined by taking derivatives of observables with respect to model parameters, but we will focus on the specific heat in the rest of our study.

3. Unlearning regularization

3.1. Unlearning algorithm

We will now describe an unsupervised algorithm employed in the realm of associative memory models that will be our main focus in the following. Inspired by the brain functioning during REM sleep [27], the Hebbian Unlearning algorithm (HU) [9–11, 27–30] is a training procedure for simple associative models. Consider a neural network having the same energy defined in equation (2). We assume the external local fields to be null, i.e. $h_i = 0$, $\forall i$. The associative memory task consists in retrieving a set of random neural states $\{\vec{\xi}^\mu\}_{\mu=1}^{\alpha N}$, called *patterns*, through a zero temperature Monte Carlo dynamics [31] initialized on some corrupted versions of them. Such a rule for the dynamics reads

$$S_i^{(t+1)} = \text{sign}\left(\sum_j J_{ij} S_j^{(t)}\right). \tag{22}$$

The control parameter α is the *load* of the model. Associativity is improved by achieving larger basins of attraction of the patterns.

The training procedure starts by initializing the connectivity matrix in the Hebb's fashion as in equation (10), i.e. $J_{ij}^0 = \langle S_i S_j \rangle_{\text{data}}$, $\forall i, j$, where $\langle \cdot \rangle_{\text{data}}$ is the empirical average over the patterns. Then, the following routine is iterated at each step t:

1. Initialize the network on a random neural state.
2. Run the zero temperature Monte Carlo dynamics, updating a randomly picked neuron at each time step according to equation (22), until convergence to a stable fixed point $\vec{S}^*$.
3. Update couplings according to:

$$\delta J_{ij}^{(t)} = -\frac{\lambda}{N} S_i^* S_j^*, \qquad J_{ii} = 0 \quad \forall i. \tag{23}$$

This algorithm was first introduced in [9] to prune the landscape of attractors from proliferating spurious states, i.e. fixed points of the neural dynamics not coinciding with the patterns [32, 33]. This pruning action stabilizes the patterns and increases the size of their basins of attraction. The numerical analysis of [11] shows that HU approaches the memory performance of a maximally stable symmetric perceptron, which is considered to be a very effective model, up to a critical capacity $\alpha_c \simeq 0.6$. Another important aspect to underline is that HU performs at its best at a given amount of algorithmic iterations that scales with N, α and λ [10, 11]. After this amount of steps the performance of the model, in terms of perfect retrieval of the patterns and size of the basins of attraction, deteriorates. This implies the necessity of applying an *early-stopping* criterion for the HU algorithm to perform optimally. We also note that HU is an *unsupervised* algorithm, in the sense that it does not need to be provided explicitly with the patterns, and only exploits the information encoded in the Hebbian initialization of the couplings.

3.2. Unlearning regularization

We now propose a new type of regularization that has the objective of imposing robustness under rescaling of the parameters, i.e.

$$\boldsymbol{J} \longrightarrow \beta \boldsymbol{J}, \qquad \boldsymbol{h} \longrightarrow \beta \boldsymbol{h}, \qquad \text{with } \beta \neq 1. \tag{24}$$

For the model to be robust under a redefinition of the parameters, we can add a regularization term to the traditional BM loss function that shifts the peak of the specific heat (i.e. the critical temperature) away from $\beta = 1$, where the data are generated. Hence, let us define the following loss function

$$\mathcal{L}(\boldsymbol{J}, \boldsymbol{h}|a) = D_{\text{KL}}(data|1) + \left(\frac{a-1}{a}\right) D_{\text{KL}}(data|a), \tag{25}$$

with

$$D_{\text{KL}}(data|\beta) = \sum_{\vec{S}} P_{\text{data}}\left(\vec{S}\right) \log\left(\frac{P_{\text{data}}\left(\vec{S}\right)}{P_\beta\left(\vec{S}\right)}\right), \tag{26}$$

being the Kullback–Leibler divergences between the data and the model inferred at an inverse temperature β. Both the cases of positive and negative regularization factor (i.e. for $a \lessgtr 1$) will be evaluated in this work. The

motivation for using the asymmetric KL distance will become clear in a few lines. The gradient descent equations for this loss function become

$$\delta J_{ij}^{(t)} = \lambda \left[a \left(\langle S_i S_j \rangle_{\text{data}} - \langle S_i S_j \rangle_a \right) - \left(\langle S_i S_j \rangle_1 - \langle S_i S_j \rangle_a \right) \right], \tag{27}$$

$$\delta h_i^{(t)} = \lambda \left[a \left(\langle S_i \rangle_{\text{data}} - \langle S_i \rangle_a \right) - \left(\langle S_i \rangle_1 - \langle S_i \rangle_a \right) \right]. \tag{28}$$

When $a = 1$, equations (27) and (28) coincide with the original BM learning. In the limit $a \to 0$ one has $\langle S_i S_j \rangle_{a \to 0} \to \langle S_i \rangle_0 \langle S_j \rangle_0 = 0$ and equation (27) tends to a thermal version of the HU rule performed at $\beta = 1$, i.e.

$$\delta J_{ij}^{(t)} = -\lambda \langle S_i S_j \rangle_1, \tag{29}$$

$$\delta h_i^{(t)} = -\lambda \langle S_i \rangle_1. \tag{30}$$

Equation (29) is similar to previous attempts in the associative memory literature [13]. On the other hand, when $a \to \infty$, and the learning rate is redefined as $\lambda = \tilde{\lambda}/a \to 0$, the algorithm becomes

$$\delta J_{ij}^{(t)} = \tilde{\lambda} \left(\langle S_i S_j \rangle_{\text{data}} - \langle S_i S_j \rangle_\infty \right), \tag{31}$$

which also resembles the unlearning procedure: the sampling at $a = \infty$ produces fixed points of the zero-temperature dynamics, which gives back equation (23). At variance with the standard routine, there is an additional Hebbian input term $\langle S_i S_j \rangle_{\text{data}}$ which impedes the couplings to vanish at convergence.

At this point, the motivation for our choice of the loss function $\mathcal{L}(\boldsymbol{J}, \boldsymbol{h}|a)$ should be clear: we search for an algorithm that interpolates between the BM learning algorithm and HU. In this way HU emerges as a particular limit of a regularization method on BMs.

4. Results

In order to benchmark our proposed regularization scheme, we first consider a small sized network, i.e. $N = \{18, 20\}$, which allows us to consider the limit of infinite number of generated data, i.e. $M \to \infty$, and compute exactly the moments $\langle \cdot \rangle_{\text{data}}$. We can then reach the fixed points of equations (27) and (28) precisely, whenever such points are admitted, with no errors due to the finite sampling. We can then compute the observables C_v or sD_{KL} more easily, in order to determine the accuracy and robustness of the inference. It must be underlined that our purpose in this framework is not to maximize the generalization properties of the model, because this goal would be trivially reached by the standard BM learning: it is rather to find the model that achieves the best compromise between accuracy and robustness. In the second instance, we generalize our analysis to the case of larger N, where quantities cannot be computed exactly.

For simplicity of notation we rename

$$\langle S_i S_j \rangle_{\text{data}} = c_{ij}^d, \qquad \langle S_i S_j \rangle_\beta = c_{ij}^\beta, \tag{32}$$

where we will mainly deal with $\beta = a$ and $\beta = 1$. The inverse temperature of the data generating model is $\beta = \beta_d$. For given β_d, we can compute c_{ij}^d exactly in the limit $M \to \infty$ by exact enumeration of the model configurations. We consider two kind of data generating models: a mean field fully connected ferromagnetic Ising network (i.e. the Curie–Weiss model) and a fully disordered spin glass network (i.e. the Sherrington–Kirkpatrick model [34]), which both display a critical point in the thermodynamic limit, manifested by a singularity of the specific heat at a given critical temperature. This singularity is smoothed in finite size systems, thus appearing as a peak in specific heat. For both models we will choose β_d to be smaller than the real position of the peak of $C_v(\beta)$ because sampling is more efficient in the paramagnetic phase.

4.1. CW model
We first consider the problem of inferring a CW model, that is defined by the following energy function

$$E_{\text{CW}} \left[\vec{S} | \boldsymbol{J} \right] = - \sum_{i,j>i} S_i J_{ij} S_j, \qquad J_{ij} = \frac{J}{N}, \forall i,j. \tag{33}$$

This model can be fully treated analytically in the finite size case, and the procedure to compute the main quantities is reported in the following. Since fields are zero, by construction of the model, we are interested in the evolution equation for the couplings, i.e.

$$\dot{J} = \lambda \left[a \left(c_d - c_a(t) \right) - \left(c_1(t) - c_a(t) \right) \right], \tag{34}$$

where we used the fact that all elements of the matrices are identical, because each node receives the same fields from its neighbors. The fixed point equation for the correlation functions from equation (34) is

$$c_a^* = \frac{c_1^*}{1-a} - \frac{a}{1-a} c_d, \tag{35}$$

where the star indicates the value of the correlations at convergence. Moreover we know that

$$Z_{\beta J} = \sum_m \binom{N}{\frac{N}{2}(1+m)} \exp\left(\frac{\beta J}{2} \left(Nm^2 - 1 \right) \right), \tag{36}$$

where $m \in [-1, -1 + \frac{2}{N}, .., 1 - \frac{2}{N}, +1]$ and we used

$$E_{\mathrm{CW}}\left[\vec{S} | J \right] = -\frac{J}{2} \left[N \left(\frac{1}{N} \sum_i S_i \right)^2 - 1 \right] = -\frac{J}{2} \left(Nm^2 - 1 \right). \tag{37}$$

From now on, we will make use of the following notation for the correlation functions: $c_\beta = c(\beta J)$, $c_d = c(\beta_d J)$. The second moment of the magnetization with respect to the Gibbs–Boltzmann measure $\langle m^2 \rangle_\beta$ can be written as

$$I_N^{(2)}(\beta J) = \frac{\sum_m m^2 \binom{N}{\frac{N}{2}(1+m)} \exp \frac{\beta}{2} J \left(Nm^2 - 1 \right)}{\sum_m \binom{N}{\frac{N}{2}(1+m)} \exp \frac{\beta}{2} J \left(Nm^2 - 1 \right)} = \frac{1 + (N-1) c(\beta J)}{N}. \tag{38}$$

Thus

$$c(\beta J) = \frac{N I_N^{(2)}(\beta J) - 1}{N - 1}. \tag{39}$$

The fourth moment of the magnetization $\langle m^4 \rangle_\beta$ can be written as

$$I_N^{(4)}(\beta J) = \frac{\sum_m m^4 \binom{N}{\frac{N}{2}(1+m)} \exp \frac{\beta}{2} J \left(Nm^2 - 1 \right)}{\sum_m \binom{N}{\frac{N}{2}(1+m)} \exp \frac{\beta}{2} J (Nm^2 - 1)} = \frac{1 + (N^3 - 1) k(\beta J)}{N^3}, \tag{40}$$

where $k(\beta J)$ is the fourth-order correlation among spins. Equation (35) then becomes

$$c(aJ) = \frac{c(J)}{1-a} - \frac{a}{1-a} c(\beta_d J), \tag{41}$$

or equivalently

$$I_N^{(2)}(aJ) = \frac{I_N^{(2)}(J)}{1-a} - \frac{a}{1-a} I_N^{(2)}(\beta_d J), \tag{42}$$

that has be to solved for J, given a and β_d.

In the following, we first choose $\beta_d = 1$ in order to generate data from the paramagnetic phase of a network of $N = 20$ spins slightly above the critical point, since the specific heat peak of the model at $N = 20$ appears to be at $\beta > 1$. Then, we solve equation (42) to obtain the coupling $J(a)$ of the inferred model in presence of the unlearning regularization. Once this is done, we analyze the resulting model by computing quantities from the Gibbs–Boltzmann distribution with parameter $\beta J(a)$ rescaled by an additional temperature β. The specific heat of the inferred model is computed as

$$C_v(\beta) = \left(\frac{\beta N J(a)}{2} \right)^2 \left(I_N^{(4)}(\beta J(a)) - I_N^{(2)\,2}(\beta J(a)) \right). \tag{43}$$

Results are reported in figure 2(a) and show that the specific heat has a peak slightly after $\beta = 1$ that shifts progressively when a is decreased as well as increased away from unity. In figure 2(b) we compare c_1 and c_a with the data correlations c_d as functions of a. There is no value of a where $c_1 = c_d$ or $c_a = c_d$, signaling that

Figure 2. Thermodynamic quantities computed for a small network trained on the configurations of the Curie–Weiss model with $N = 20$ spins at a given β_d. (a) The specific heat of the inferred model for $\beta_d = 1$ is reported as a function of the inverse temperature β, at different values of the regularization parameter a. Recall that $a = 1$ corresponds to standard BM learning that reproduces the original model exactly. (b) A comparison between c_1, c_a and the empirical data correlation c_d when inferred at different values of a. The only point where $c_1 = c_a = c_d$, i.e. $a = 1$, is marked by a red circle. (c) Inferred coupling multiplied by a/β_d as a function of a at different values of β_d. The point $a = 1$ where $c_1 = c_a = c_d$ is marked by a red circle.

the inferred model never reproduces exactly the original system, except for the trivial case of $a = 1$. Moreover, c_1/c_d and c_a/c_d swap when crossing $a = 1$, as one could guess by the fact that the regularization factor in equation (25) passes from being negative to positive. Furthermore, in figure 2(c) we compute $J(a)$ for different values of β_d and plot the quantity $aJ(a)/\beta_d$ as a function of a: when a lies in between 0 and 1 the lines are approximately linear, signaling that $J(a)$ is nearly constant in a. The inferred model thus changes slowly upon varying a and this property improves when N is increased.

4.2. SK model: training

Next, we consider the problem of inferring a SK model, i.e.

$$E_{\text{SK}}\left[\vec{S}|\boldsymbol{J}\right] = -\sum_{i,j>i} S_i J_{ij} S_j, \quad J_{ij} \sim \mathcal{N}\left(0, N^{-1/2}\right). \tag{44}$$

We consider a network of $N = 18$ spins with a data generation temperature $\beta_d = 0.4$. For clarity of the results, we will use only one realization of the parameters for the SK, having a peak in the specific heat at a β slightly larger than β_d. The parameters are initialized as $\boldsymbol{J}^{(0)} = c_d$ and $\boldsymbol{h}^{(0)} = \boldsymbol{0} \forall i$.

The fixed points of the gradient descent equations (27) and (28) is such that the inferred model does not exactly reproduce the original system, due to the regularization. In fact, when $a \ll 1$ the stable fixed point of the equations is $\boldsymbol{c}_1 \simeq a\boldsymbol{c}_d$ and $\boldsymbol{c}_a \simeq 0$; when $a \gg 1$, one has $\boldsymbol{c}_a = \boldsymbol{c}_d$ and $\boldsymbol{c}_1 \simeq 0$. In both scenarios the inferred model (at $a = 1$) does not show a good accuracy. Note that even if for $a \gg 1$ the model scaled by a fits well the data, we do not consider this as a good inferred model because it is not robust upon further lowering the temperature.

For this reason, we also consider an *early-stopping* criterion for the training, such that the unlearning regularization reaches the best compromise between accuracy and robustness. Let us compute the symmetric KL divergence between the original SK model and the inferred one at different number of training steps. Figure 3(a) shows the resulting curve for a model regularized via the unlearning scheme with $a = 0.3$ and compares the standard Hebbian initialization of the couplings with two other initializations: $J_{ij}^{(0)} = 0$ and $J_{ij}^{(0)} \sim \mathcal{N}(0, N^{-1/2}) \forall i, j$. All the three curves reach the same final performance, by convexity of the loss function. Nevertheless, only the Hebbian initialization displays a local minimum in sD_{KL} during the initial transient, which signals a high resemblance of the inferred network to the data-generating model.

To compare the correlation matrices $\boldsymbol{c}_1$ and $\boldsymbol{c}_d$, a non-weighted fit of the elements of c_d versus the corresponding elements of c_1 provides the slope $\omega(\boldsymbol{c}_1, \boldsymbol{c}_d)$, while the Pearson coefficient $\rho(\boldsymbol{c}_1, \boldsymbol{c}_d)$ measures the degree of correlation between the two matrices. A high accuracy is reached when both ρ and ω are close to unity. Note that the estimation of ω and ρ carry the same computational cost, in fact at each algorithm time step we have

$$\omega^{(t)}\left(\boldsymbol{c}_1, \boldsymbol{c}_d\right) = \frac{\rho^{(t)}\left(\boldsymbol{c}_1, \boldsymbol{c}_d\right) - \mu_d \cdot \mu_1(t)}{s_1^2(t) - \mu_1(t)^2}, \tag{45}$$
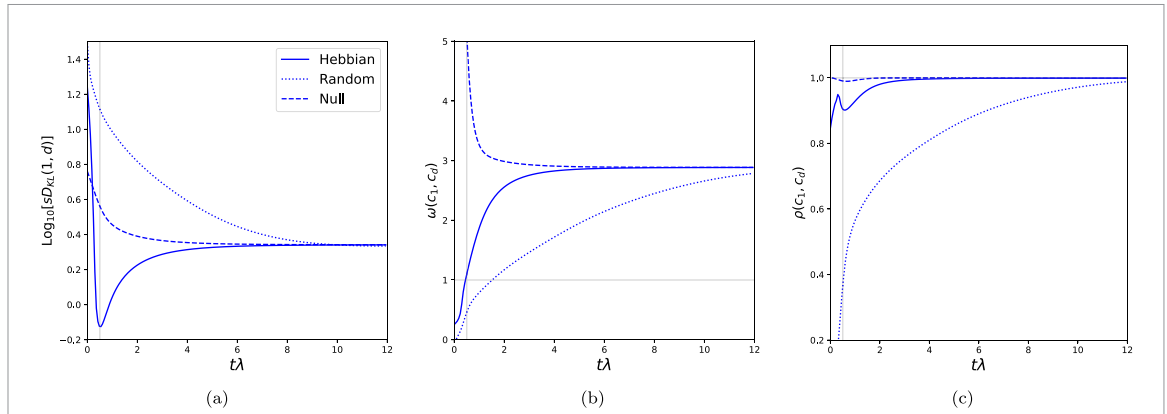
Figure 3. (a) Symmetric KL divergence (in log scale), as a function of time $t\lambda$, between the inferred Gibbs–Boltzmann distribution at unitary temperature and the one describing a SK model at $\beta_d = 0.4$. (b) Slope ω obtained from the linear fit of the elements of c_d versus the elements of c_1, as a function of time t. (c) Pearson coefficient ρ between the correlation matrix c_1 and the data correlation matrix c_d as a function of time t. The *full* line reports the case of Hebbian initialization of the couplings; the *dotted* line corresponds to a random initialization; the *dashed* line represents the initialization to $J^{(0)} = 0$. The gray vertical line signs the location of the minimum of sD_{KL} for the Hebbian initialization. Choice of the parameters: $N = 18$, $a = 0.3$, $\lambda = 0.06$.
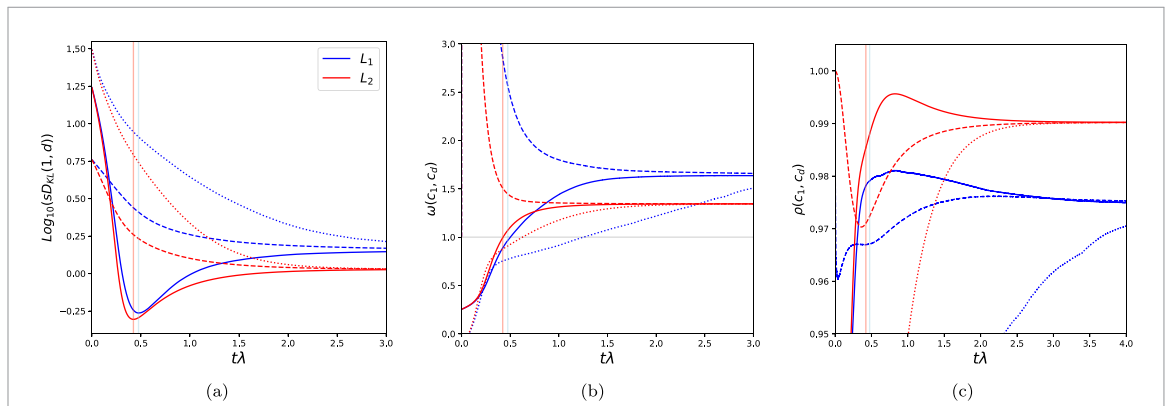

Figure 4. (a) Symmetric KL divergence (in log scale), as a function of time $t\lambda$, between the Gibbs–Boltzmann distribution at unitary temperature, inferred via L_1 and L_2 regularizations, and the one describing a SK model at $\beta_d = 0.4$. (b) Slope ω obtained from the linear fit of the elements of c_d versus the elements of c_1, as a function of time $t\lambda$. (c) Pearson coefficient ρ between the correlation matrix c_1 and the data correlation matrix c_d as a function of time $t\lambda$. The *full* line reports the case of Hebbian initialization of the couplings; the *dotted* line corresponds to a random initialization; the *dashed* line represents the initialization to $J^{(0)} = 0$. Light blue and red vertical lines sign the locations of the minima of sD_{KL} for the Hebbian initialization, respectively in the L_1 and L_2 methods. Choice of the parameters: $N = 18$, $\beta_d = 0.4$, $\lambda_1 = \lambda_2 = 0.001$, $\gamma_1 = 0.15$, $\gamma_2 = 1$.

where μ_d is the average over the off-diagonal elements of c_d and $\mu_1(t), s_1^2(t)$ are, respectively, the average of the off-diagonal elements and the average of the off-diagonal squared elements of c_1 at time t.

Figures 3(b) and (c) report the evolution of ω and ρ during training. As one can notice, there is a clear correspondence between the local minimum of the symmetric KL divergence and the point where $\omega = 1$ and ρ approaches a local maximum. It is therefore interesting to consider stopping the training at this point (defined in practice by $\omega = 1$), in order to achieve the best accuracy. We note that this is only possible with a Hebbian initialization of the parameters; while a null initialization never reaches $\omega = 1$, a random one reaches it but with a much lower Pearson coefficient ρ, resulting in poor accuracy. Note that the correspondence between the local minimum of sD_{KL} and the best match of c_d and c_1 is not valid for the entire range of a. Specifically, for the rest of our analysis we will avoid values of a that are close to unity, because for this value the correspondence does not hold and we cannot define a good early-stopping criterion.

We also observe that L_p regularizations do not converge to the best network in terms of accuracy: even though $\rho \simeq 1$ at convergence, we find $\omega \neq 1$. Nevertheless, we found that L_1 and L_2 regularization schemes with a Hebbian initialization also display the same correspondence between a minimum in sD_{KL} and the point where ω approaches unity and ρ admits a local maximum in time, as shown in figure 4.
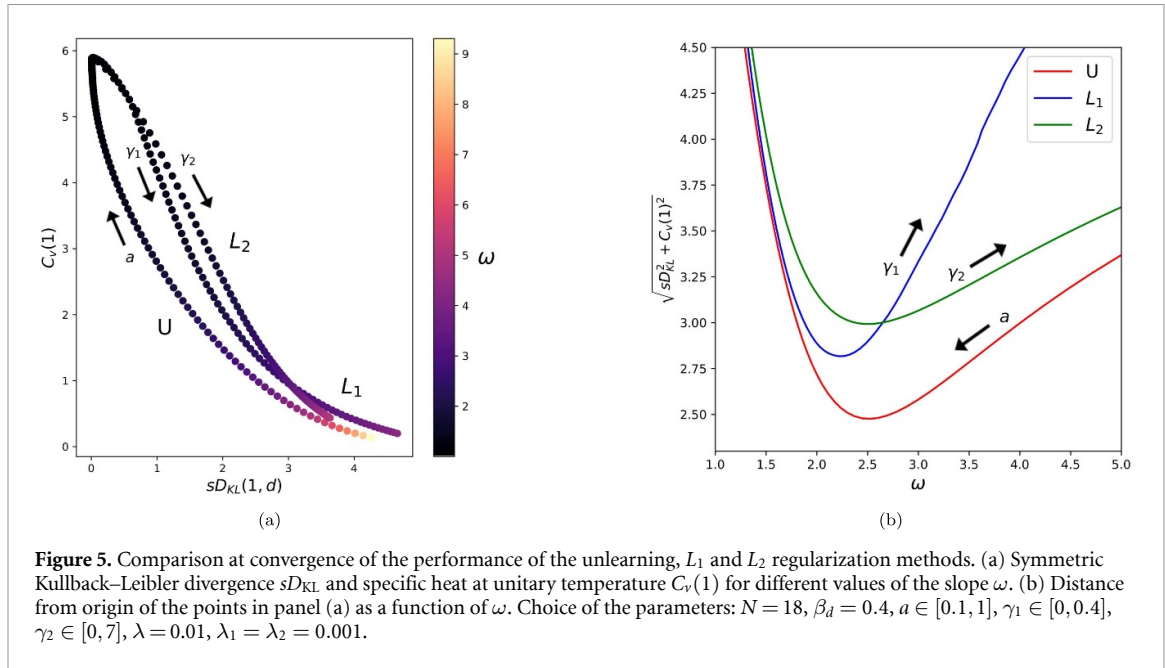
Figure 5. Comparison at convergence of the performance of the unlearning, L_1 and L_2 regularization methods. (a) Symmetric Kullback–Leibler divergence sD_{KL} and specific heat at unitary temperature $C_v(1)$ for different values of the slope ω. (b) Distance from origin of the points in panel (a) as a function of ω. Choice of the parameters: $N = 18$, $\beta_d = 0.4$, $a \in [0.1, 1]$, $\gamma_1 \in [0, 0.4]$, $\gamma_2 \in [0, 7]$, $\lambda = 0.01$, $\lambda_1 = \lambda_2 = 0.001$.

4.3. SK model: results at convergence

Before discussing the early-stopping results, we now compare the generalization performance of the unlearning regularization with the L_1 and L_2 regularizations described in section 2.2, at convergence. Data are generated by the same SK model with $\beta_d = 0.4$, as in section 4.2.

The couplings are optimized until convergence of the training for the three types of regularizations. Three quantities are measured after training: the symmetric Kullback–Leibler divergence $sD_{\text{KL}}(1, d)$ between the model at unitary temperature and the original SK model, the specific heat at unitary temperature $C_v(1)$ and the slope ω defined as in equation (45). One should keep in mind that $\omega = 1$ is reached only by the trivial choice of the regularization parameters $a = 1$ and $\gamma_1 = 0$, $\gamma_2 = 0$; any other choice of these parameters leads to a reduced accuracy, by definition of regularization. Measures are repeated for different choices of the regularization parameters $\{a, \gamma_1, \gamma_2\}$ and results are reported in figure 5. Specifically, figure 5(a) is a color plot showing the performance of the network in the plane $(sD_{\text{KL}}, C_v(1))$, where each point is colored according to the slope ω. The three lines correspond to the three regularizations: the unlearning line is generally closer to the origin than the L_1 and L_2 ones, suggesting a better compromise between accuracy and robustness. All the lines converge to the ground-truth when $sD_{\text{KL}} = 0$, corresponding to absence of regularization ($a = 1$ or $\gamma_1 = 0$ or $\gamma_2 = 0$).

To better compare the three regularization methods, figure 5(b) reports the distance of the points reported in figure 5(a) from the origin directly as a function of the slope ω. Consistently with figure 5(a), the plot shows that the set of points relative to unlearning always stands below the L_1 and L_2 ones, while all regularizations tend to the ground truth when ω approaches unity. Different experiments with different data and temperatures β_d show very similar results.

One can repeat the analysis by comparing the three regularization schemes at constant standard deviation of the couplings σ_J instead of constant slope ω. The results are reported in figure 6, and the unlearning regularization once again reaches a better compromise between accuracy and robustness compared to the other regularization methods. Note that the U and L_2 curves tend to coincide when $\sigma_J \to 0$, i.e. when all inferred couplings are small. This can be explained through a weak coupling expansion of the gradient equations for U and L_2. When $\sigma_J \ll 1$ the two-point correlation functions can be Taylor expanded as

$$c_{ij}^{\beta}(t) = \beta J_{ij}^{(t)} + \mathcal{O}\left(J_{ij}^{(t)^2}\right). \tag{46}$$

The expansion of equation (27) leads to

$$\delta \boldsymbol{J}^{(t)} = \lambda \left(a \boldsymbol{c}_d - (1 - a) \boldsymbol{J}^{(t)} + \mathcal{O}\left(a^2\right)\right). \tag{47}$$

Since σ_J is small when $a \ll 1$, in this limit we obtain

$$\delta \boldsymbol{J}^{(t)} \simeq \lambda \left(a \boldsymbol{c}_d - \boldsymbol{J}^{(t)}\right), \tag{48}$$

Figure 6. Comparison at convergence of the performance of the unlearning, L_1 and L_2 regularization methods. (a) Symmetric Kullback–Leibler divergence sD_{KL} and specific heat at unitary temperature $C_v(1)$ for different values of the standard deviations of the couplings σ_J. (b) Distance from origin of the points in panel (a) as a function of σ_J. Choice of the parameters: $N = 18$, $\beta_d = 0.4$, $a \in [0.1, 1]$, $\gamma_1 \in [0, 0.4]$, $\gamma_2 \in [0, 7]$, $\lambda = 0.01$, $\lambda_1 = \lambda_2 = 0.001$.

which at the fixed point becomes

$$\boldsymbol{J} = a\boldsymbol{c}_d \longrightarrow \sigma_J = a\sigma_d, \tag{49}$$

where σ_d is the standard deviation of the empirical correlation matrix. By repeating the same procedure with equation (18) one obtains

$$\delta\boldsymbol{J}^{(t)} = \lambda\left(\boldsymbol{c}_d + (1 - \gamma_2)\boldsymbol{J}^{(t)}\right), \tag{50}$$

which at the fixed point becomes

$$\boldsymbol{J} = \frac{\boldsymbol{c}_d}{\gamma_2 - 1} \longrightarrow \sigma_J = \frac{\sigma_d}{\gamma_2 - 1}, \tag{51}$$

for $\gamma_2 > 1$, which is justified by the fact that σ_J is small when γ_2 is sufficiently large. Numerics support the fact that, when $\sigma_J \ll 1$ and σ_J is the same in figure 5 for both U and L_2, then $a(\gamma_2 - 1) = 1$. Our results show that the HU algorithm, which derives from the unlearning regularization when $a \to 0$, can also be obtained via a L_2 regularization with a large γ_2.
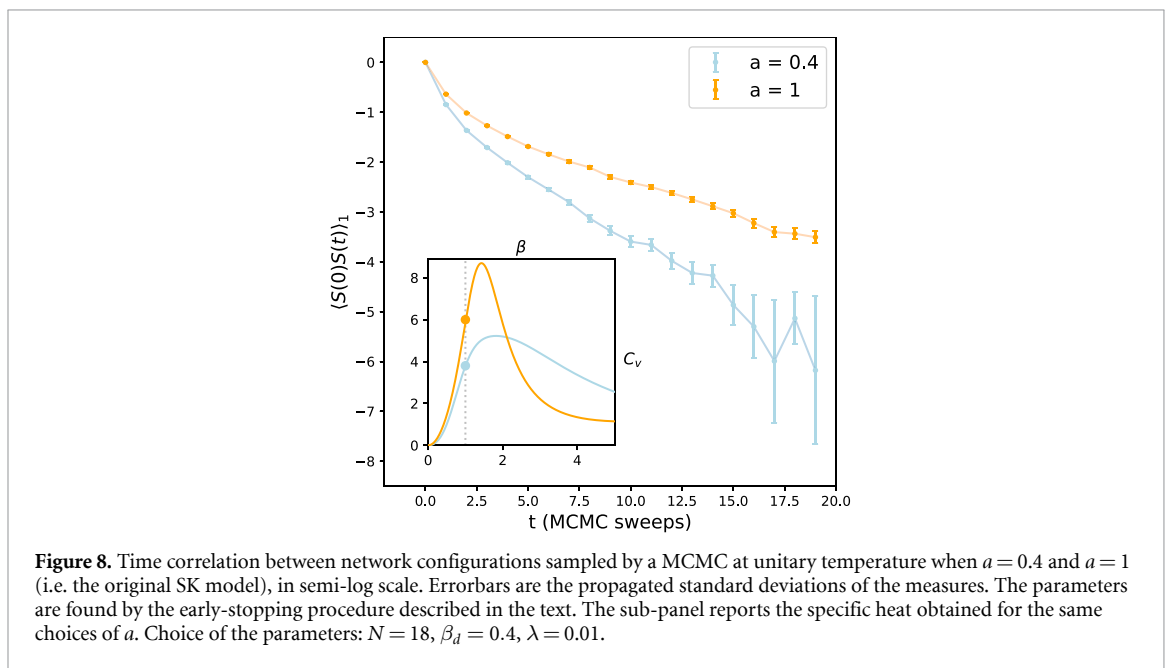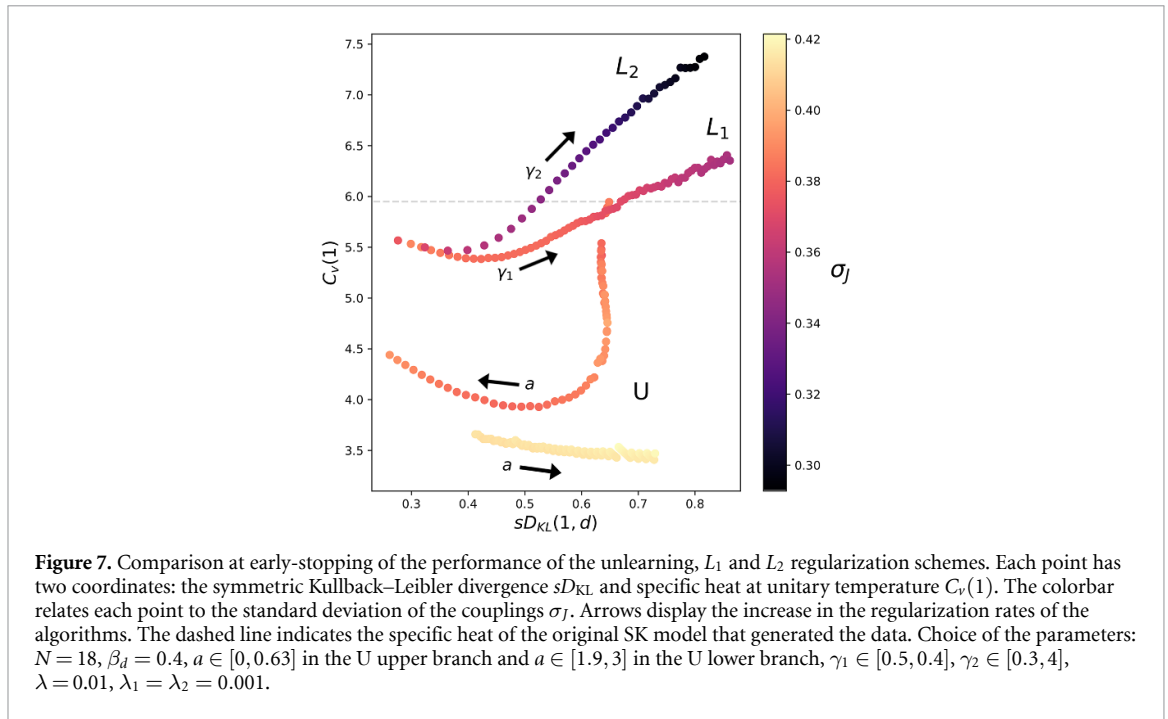
4.4. SK model with early-stopping

The early-stopping criterion described in section 4.2 is now employed to improve the performance of the inferred model. The training is stopped when $\omega \simeq 1$. For the unlearning regularization, we consider a range of values for a such that there is a good correspondence among the minimum of sD_{KL} and ω, ρ reaching unity: outside the considered interval these three quantities might not behave consistently, and we could not define a good early-stopping criterion. Measures of the symmetric Kullback–Leibler divergence $sD_{KL}(1, d)$, the specific heat at unitary temperature $C_v(1)$ and the standard deviation of the couplings σ_J are repeated for different choices of the regularization parameters $\{a, \gamma_1, \gamma_2\}$ and results are reported in figure 7 by a color plot showing the performance of the network in the plane $(sD_{KL}, C_v(1))$, where each point is colored according to the standard deviation σ_J.

The specific heat of the original SK model is reported as a dashed line in the plot. The figure shows three lines relative to the three regularizations: at equal value of sD_{KL}, the unlearning regularization has much lower $C_v(1)$ than the L_1 and L_2 regularizations, signaling a higher robustness at comparable accuracy. We observe a range of values of γ_1 and γ_2 for which the L_1 and L_2 regularizations reduce the specific heat in $\beta = 1$, while the remaining choices for the regularizers do not allow to obtain a higher robustness with respect to the original SK model, i.e. $C_v(1)$ is increased. Different calculations with different data and temperatures β_d produce very similar results.

Finally, we show that the unlearning regularization helps generating new data. Since the inferred model is less critical than the original one, the correlation between network configurations sampled at different time

Figure 7. Comparison at early-stopping of the performance of the unlearning, L_1 and L_2 regularization schemes. Each point has two coordinates: the symmetric Kullback–Leibler divergence sD_{KL} and specific heat at unitary temperature $C_v(1)$. The colorbar relates each point to the standard deviation of the couplings σ_J. Arrows display the increase in the regularization rates of the algorithms. The dashed line indicates the specific heat of the original SK model that generated the data. Choice of the parameters: $N = 18$, $\beta_d = 0.4$, $a \in [0, 0.63]$ in the U upper branch and $a \in [1.9, 3]$ in the U lower branch, $\gamma_1 \in [0.5, 0.4]$, $\gamma_2 \in [0.3, 4]$, $\lambda = 0.01$, $\lambda_1 = \lambda_2 = 0.001$.

Figure 8. Time correlation between network configurations sampled by a MCMC at unitary temperature when $a = 0.4$ and $a = 1$ (i.e. the original SK model), in semi-log scale. Errorbars are the propagated standard deviations of the measures. The parameters are found by the early-stopping procedure described in the text. The sub-panel reports the specific heat obtained for the same choices of a. Choice of the parameters: $N = 18$, $\beta_d = 0.4$, $\lambda = 0.01$.

steps of a Monte Carlo chain at unitary temperature are smaller. Figure 8 shows a comparison between the time-correlation functions corresponding to the inferred system at $a = 0.4$ and the original SK model. The inset displays the behavior of the specific heat C_v for the same two models.

4.5. SK model with $N \gg 1$

In this section, we test the method on an instance with larger number N of variables, showing a good efficiency of the Unlearning method, both run until convergence and early-stopped.

A number $M = 10^4$ of data-points are sampled from a SK model with $N = 200$ spins in equilibrium at a temperature $\beta_d = 0.4$ (i.e. in the paramagnetic phase). The network is then initialized in Hebbian fashion, i.e. $\boldsymbol{J}^{(0)} = \boldsymbol{c}_d$ and $\boldsymbol{h}^{(0)} = \boldsymbol{0}$. The model is trained via Unlearning, L_1 and L_2 regularization methods and the specific heat $C_v(1)$, the slope $\omega(\boldsymbol{c}_1, \boldsymbol{c}_d)$ and the Pearson coefficient $\rho(\boldsymbol{c}_1, \boldsymbol{c}_d)$ are measured through MCMC sampling at unitary temperature. Two different training prescriptions have been adopted for the study: the algorithm is executed until convergence of equations (27), (28); training is early-stopped as soon as $\omega \simeq 1$.
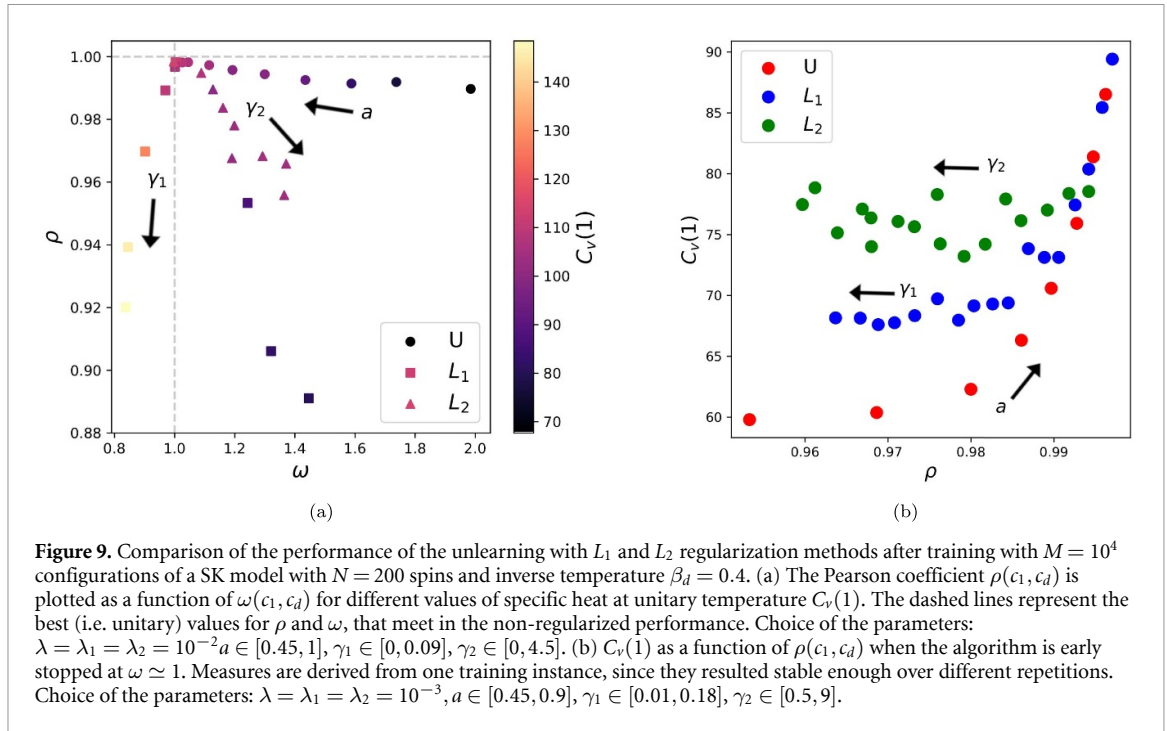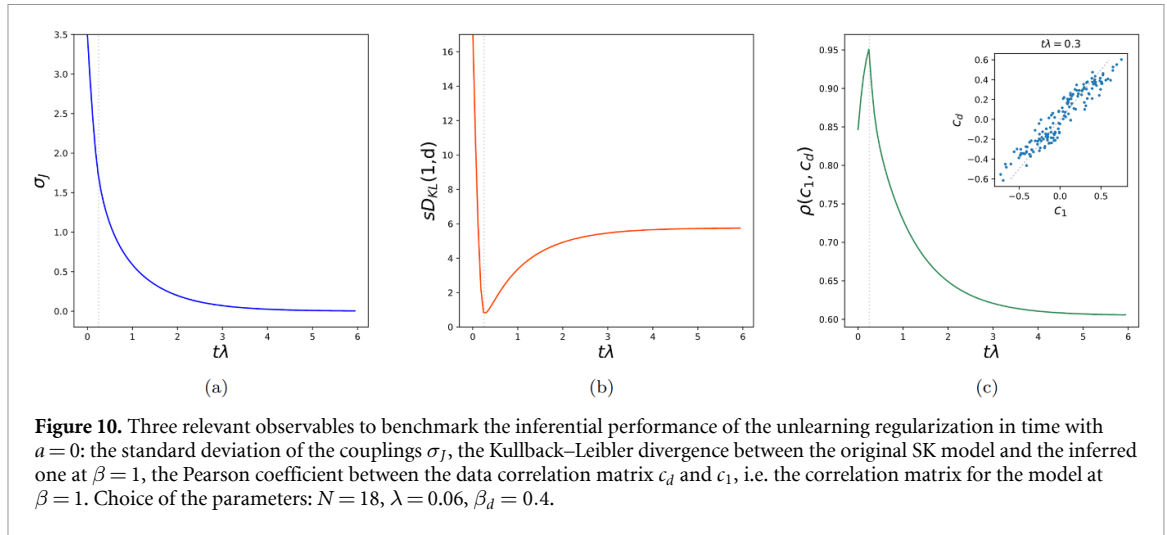
Figure 9. Comparison of the performance of the unlearning with L_1 and L_2 regularization methods after training with $M = 10^4$ configurations of a SK model with $N = 200$ spins and inverse temperature $\beta_d = 0.4$. (a) The Pearson coefficient $\rho(c_1, c_d)$ is plotted as a function of $\omega(c_1, c_d)$ for different values of specific heat at unitary temperature $C_v(1)$. The dashed lines represent the best (i.e. unitary) values for ρ and ω, that meet in the non-regularized performance. Choice of the parameters: $\lambda = \lambda_1 = \lambda_2 = 10^{-2} a \in [0.45, 1], \gamma_1 \in [0, 0.09], \gamma_2 \in [0, 4.5]$. (b) $C_v(1)$ as a function of $\rho(c_1, c_d)$ when the algorithm is early stopped at $\omega \simeq 1$. Measures are derived from one training instance, since they resulted stable enough over different repetitions. Choice of the parameters: $\lambda = \lambda_1 = \lambda_2 = 10^{-3}, a \in [0.45, 0.9], \gamma_1 \in [0.01, 0.18], \gamma_2 \in [0.5, 9]$.

Figure 9 depicts the values of ρ as a function of ω and $C_v(1)$ as achieved through the three regularization methods for various values of the control parameters a, γ_1, γ_2. At convergence (see figure 9(a), the Unlearning algorithm appears to approach a better accuracy performance (i.e. in terms of both ρ and ω) together with a lower value of the specific heat. Even after early-stopping (see figure 9(b) the same behavior emerges: there is a range in a where, given the same degree of accuracy, the model achieves a smaller specific heat with respect of the other regularization methods. We also stress that, by contrast with the rest of our analysis, this experiment provides for a specific heat $C_v(1)$ obtained at early-stopping which is lower than the one reached at convergence. This aspect might be due to the Hebbian initialization of the parameters in a regime which is very far from the perfect-sampling one that we considered before.

It should be noted that the standard deviation σ_J of the couplings is chosen to have the same order of magnitude, across all three regularization methods. Both in the converged and early-stopped scenario, the Unlearning algorithm is superior to the L_p regularizations, confirming the results obtained at small N (i.e. the perfect sampling framework). Moreover, the early-stopping technique used in this work, which relies on the Hebbian initialization of the network, results handily applicable for larger dimensions due to its reduced computational cost, hence providing an effective training recipe in terms of both accuracy and robustness.

5. The Hebbian unlearning limit

We will now study the specific limit of the unlearning regularization that leads to an algorithm which strongly resembles the HU routine. The inferential performance of the learning procedure is examined by measuring the distance between the inferred model and the original one at different training steps. Results show, for the first time, that HU can be employed as an inferential tool, due to the beneficial effect of the Hebbian initialization of the couplings. We advance an explanation for its performance that is supported by further numerical evidence. In the limit $a \to 0$ the updating rules for the parameters transform into equations (29) and (30). As mentioned before, equation (29) strongly resembles the traditional HU algorithm of [9]. By contrast with the original rule, the new regularization samples configurations at $\beta = 1$ instead of stable fixed points of the neural dynamics, and makes use of a thermal average, rather than summing each contribution each time. We will keep dealing with small networks so that, given the initial conditions for the parameters, the loss function of the problem can be minimized exactly. As a numerical experiment, we train a BM with $N = 18$ neurons to learn a SK model at $\beta_d = 0.4$. The unlearning regularization is applied with $a = 0$ and the usual initial conditions for the parameters, i.e. $\boldsymbol{J}^{(0)} = \boldsymbol{c}_d$ and $\boldsymbol{h}^{(0)} = \boldsymbol{0}$.

Even if training is performed over both the couplings and the fields, according to the rules (29) and (30), the fields $\boldsymbol{h}$ do not appear in the energy of the original model, hence we will focus exclusively on the evolution of the interactions $\boldsymbol{J}$. Figure 10(a) displays the standard deviation of the couplings: as we can see the general trend shows an exponential decay of the interactions, as one typically observes in the HU

Figure 10. Three relevant observables to benchmark the inferential performance of the unlearning regularization in time with $a = 0$: the standard deviation of the couplings σ_J, the Kullback–Leibler divergence between the original SK model and the inferred one at $\beta = 1$, the Pearson coefficient between the data correlation matrix c_d and c_1, i.e. the correlation matrix for the model at $\beta = 1$. Choice of the parameters: $N = 18$, $\lambda = 0.06$, $\beta_d = 0.4$.

algorithm [10]. The decay to zero is implied by the fact that the fixed point of the gradient descent equations for equation (29) corresponds to $c_1 \equiv 0$. Figure 10(b) depicts the symmetric Kullback–Leibler divergence between the Gibbs–Boltzmann distribution of the original SK model, from which data were sampled, and the inferred model with $\beta = 1$. The sD_{KL} is high at the beginning, then decreases until reaching a global minimum signaling an optimum in the inferential performance, then it increases again and stabilizes at a plateau. The minimum is sufficiently low to indicate a good statistical consistency between the model and data. The Pearson coefficient between the correlation matrix of the data c_d and the one of the inferred model at unitary temperature c_1 is measured step-by-step in the learning process and reported in figure 10(c): as one can notice, there is a global maximum near the position of the global minimum of the sD_{KL}. The inset in figure 10(c) displays the good agreement between the two matrices, the inferred c_1 and c_d, at the position of the peak.

This analysis suggests that the unlearning regularization with $a = 0$ displays two algorithmic regimes: a transient regime where the model at $\beta = 1$ shows a good statistical agreement with the generating model; another regime where $c_1 \to 0$, and the total performance deteriorates. The first transient regime is the most important one, because it suggests that HU can be considered as an inferential tool, aside of its associative memory use. We know from section 2.1 that in BM learning data are shown to the model each time-step of the algorithm and this imposes the moment matching. The co-existence of a positive Hebbian term and a negative unlearning one in the traditional BM learning has already been pointed out in [1, 35]. In fact, the variation of each pair of couplings J_{ij} is given by

$$\delta J_{ij} = \delta^H J_{ij} + \delta^U J_{ij}, \tag{52}$$

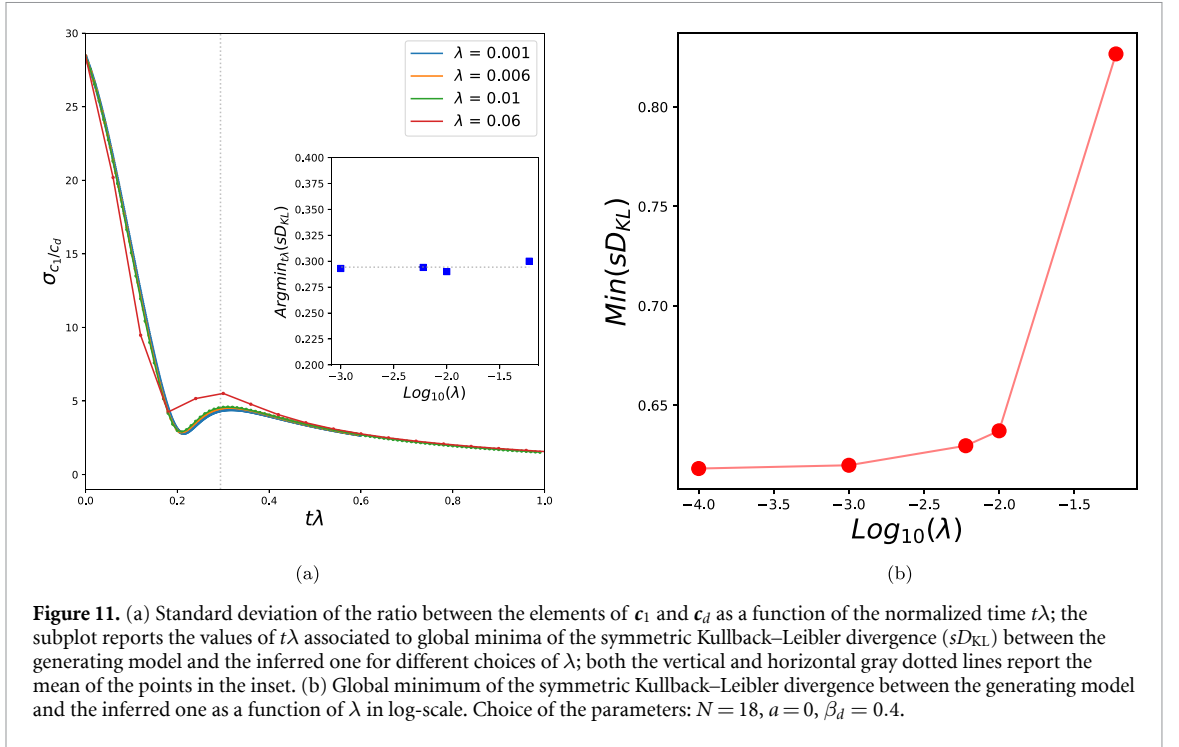where we neglected the dependence on time, and

$$\delta^H J_{ij} = \langle S_i S_j \rangle_{\mathrm{data}}, \tag{53}$$

$$\delta^U J_{ij} = -\langle S_i S_j \rangle_{\mathrm{mod}}. \tag{54}$$

We can interpret each step in the minimization of $\mathcal{L}$ as the result of two contributions, one constant quantity deriving from the data, and another one deriving from the evolving model. While a standard BM can start wherever in the space of the parameters and ends up at the fixed point of equation (52), in the HU case the update is performed by using only the negative unlearning contribution. Nevertheless, the data have been seen by the model, specifically through the Hebbian choice of the initial conditions. This observation suggests that HU approaches the minimum of the loss function in two temporally separated steps: by first descending along the *data* direction and then, progressively, along the *model* one. As a consequence, we can assume that such a two-step minimization is similar to standard BM learning as long as

$$|\langle S_i S_j \rangle_{\mathrm{mod}}| \gg |\langle S_i S_j \rangle_{\mathrm{data}}| \tag{55}$$

for most of the pairs i, j. To test whether this condition holds while training a BM regularized with $a = 0$, and initialized with $J^{(0)} = c_d$, we measure the standard deviation of the ratio between the elements of c_1 and c_d as a function of time.

Figure 11. (a) Standard deviation of the ratio between the elements of c_1 and c_d as a function of the normalized time $t\lambda$; the subplot reports the values of $t\lambda$ associated to global minima of the symmetric Kullback–Leibler divergence (sD_{KL}) between the generating model and the inferred one for different choices of λ; both the vertical and horizontal gray dotted lines report the mean of the points in the inset. (b) Global minimum of the symmetric Kullback–Leibler divergence between the generating model and the inferred one as a function of λ in log-scale. Choice of the parameters: $N = 18$, $a = 0$, $\beta_d = 0.4$.
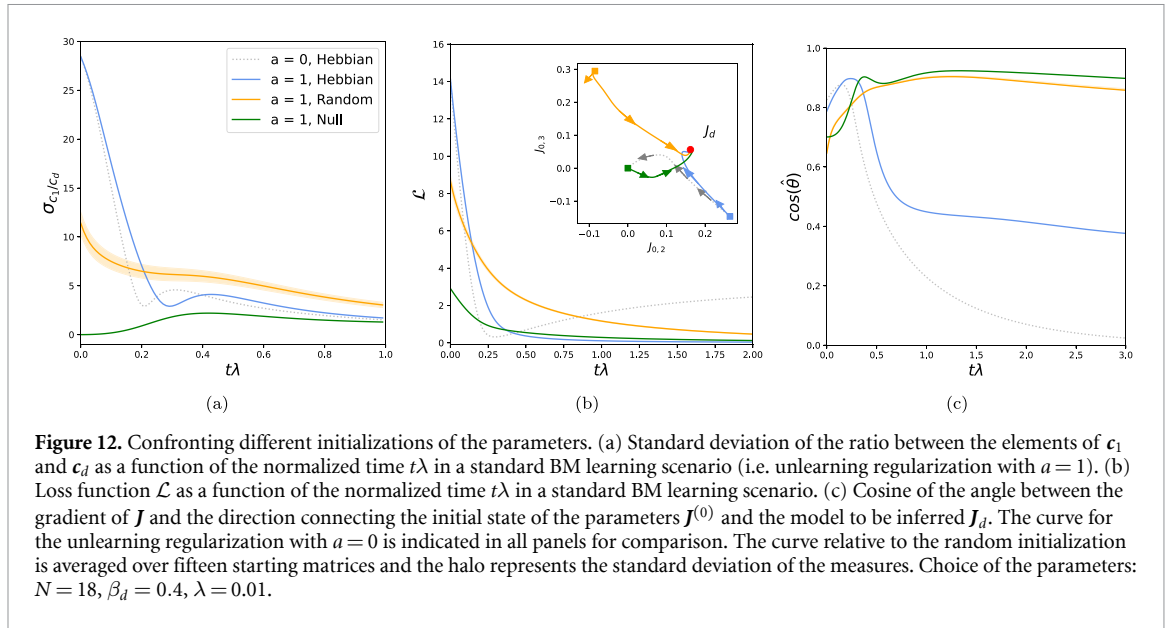
Results are presented in figure 11(a) for the usual data-set sampled by a SK model. The standard deviation of the ratio of the elements of the two matrices c_1 and c_d is plotted as a function of $t\lambda$ for different choices of λ. The curves collapse well from λ smaller than $\mathcal{O}(10^{-2})$. The system starts with c_1 generally being much larger than c_d, which satisfies the condition (55). Around $t \sim 0.2\lambda^{-1}$ the curves reach a local minimum, then increase until a local maximum at $t \sim 0.29\lambda^{-1}$, to start a slow decay right after. Both these two stationary points are located where c_1 and c_d share the same order of magnitude, i.e. where the condition for the two-steps minimization of the loss ceases to hold. However, only the second stationary point is associated to the minimum of the sD_{KL} between the original SK and the model, as reported in the inset of figure 11(a). Moreover, figure 11(b) reports the values assumed by sD_{KL} at its global minimum for various choices of λ. As one can notice, the generalization of the model increases when λ is small: we can imagine that the smaller is λ the stronger is the effect of the initial overshoot along the *data* direction.

Different parameter initializations can now be compared. Specifically, we run the standard BM learning (i.e. the unlearning regularization with $a = 1$) after a Hebbian, random, and all-to-zero initialization of the couplings. The Hebbian and the random initializations have the same initial standard deviation of the couplings σ_J. At this stage we measure the evolution in time of the standard deviation of the ratio c_1/c_d and the value of the loss. In addition to these two quantities, we introduce another observable that quantifies the degree of alignment of the gradient $\vec{\delta J}(t) = -\vec{\nabla}\mathcal{L}(t)$ to the direction connecting the initial state of the couplings, i.e. $J^{(0)}$, and the convergence state $J_d = \beta_d J^{SK}$. Such quantity is defined as

$$\cos\left(\hat{\theta}\right) = \frac{\vec{\delta J} \cdot \left(\vec{J_d} - \vec{J}^{(0)}\right)}{|\vec{\delta J}||\vec{J_d} - \vec{J}^{(0)}|}. \tag{56}$$

The evolution of these observables are reported in figure 12. Note that, for the random initialization, measures are averaged over fifteen realizations of the initial matrix. As one can observe from figure 12(a), the highest dispersion of the elements of c_1/c_d is obtained through the Hebbian initialization; the random initialization also begins with the elements of c_1 being on average larger than the empirical data correlations. Yet, as one can see from figure 12(b), the loss decays generally faster in the Hebbian case, with respect to the random case. Figure 12(c) allows to interpret the behavior of the learning dynamics in terms of the gradient descent: the Hebbian initialization points the gradient $\vec{\delta J}$ towards the minimum of the loss, located in J_d; on the other hand, the random initialization does not start well in its descent, and progressively adjusts the trajectory. The wrong start of training after random initialization can be also visualized in the inset of figure 12(b), displaying a projection of the trajectory over the first two components of the J matrix, with arrows indicating the direction of the gradient descent. Only one experiment is reported for the random initialization since the others showed very similar behavior. Conversely, when the system learns from zero
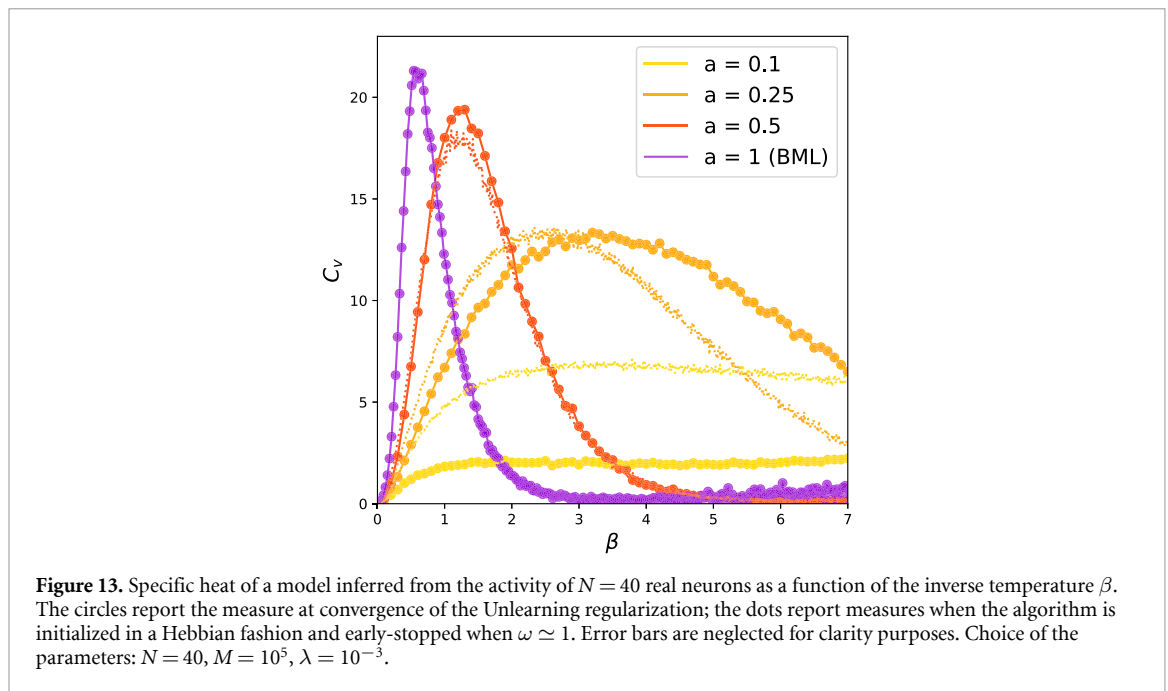
Figure 12. Confronting different initializations of the parameters. (a) Standard deviation of the ratio between the elements of c_1 and c_d as a function of the normalized time $t\lambda$ in a standard BM learning scenario (i.e. unlearning regularization with $a = 1$). (b) Loss function $\mathcal{L}$ as a function of the normalized time $t\lambda$ in a standard BM learning scenario. (c) Cosine of the angle between the gradient of J and the direction connecting the initial state of the parameters $J^{(0)}$ and the model to be inferred J_d. The curve for the unlearning regularization with $a = 0$ is indicated in all panels for comparison. The curve relative to the random initialization is averaged over fifteen starting matrices and the halo represents the standard deviation of the measures. Choice of the parameters: $N = 18$, $\beta_d = 0.4$, $\lambda = 0.01$.

initial couplings, both the positive and negative contributions to the learning appear to be small and comparable in value, which implies that the implementation of pure unlearning is ineffective in this case; the descent of the loss is slower with respect to the Hebbian initialization; the trajectory is already sufficiently pointing to the bottom of the landscape, and progressively adjusts itself.

We conclude by stating another important remark. The unlearning trajectory and the standard BM learning look very similar in the first stage of training, in terms of the gradient descent, as predicted by equation (55). After a certain amount of iterations that we have identified above, the unlearning trajectory deviates towards $J = 0$, while the BM finally converges to the final target. The abrupt change of direction that is visible from both the two-dimensional projection of the trajectory and the evolution of the gradient might be related to the choice of the data-set. We have thus showed that, when the network is initialized in the Hebbian fashion, the dominant contribution to the BM learning is the unlearning one, and the gradient of J points towards the minimum of the loss function. Hence we showed that the HU algorithm works as well as an inferential device than as a training procedure for associative memory models.

6. A real-world application

This section is dedicated to the testing of the performance of the Unlearning regularization on real data. We train a BM on a data-set available from [16], reporting the activity of $N = 40$ retinal ganglion neurons reacting to visual stimuli. We firstly binarize each data vector by discretizing a time interval of $0.1s$ into bins $\Delta\tau = 20$ ms, as done in [7, 16]. As a consequence, each data-point is a sequence of S_i units, representing the instance where the ith neuron has fired ($S_i = +1$) or not fired ($S_i = -1$) in given time bin. Each time bin is thus associated to a data-point of our binary data-set. We are provided with several repetitions of the experiment over the same population of neurons, so that we can train the model with $M = 10^5$ data-points in total.

The model is trained as a BM and regularized over couplings and fields through the Unlearning method with different choices of the parameter a. The profile of the specific heat C_v is measured as a function of β and reported in figure 13. Lines with dots represent a machine trained until convergence of the algorithm. The curve obtained at $a = 1$, corresponding to pure BM learning, appears consistent with the one reported in [7]. For each $a < 1$ the peak of the specific heat shifts away from unit, and its value in $\beta = 1$ is lowered by the regularization, suggesting a higher stability of the inferred network under a rescaling of couplings and fields. The small dots report the specific heat measured under the usual early-stopped prescription when couplings and fields are initialized as in equations (10) and (11): even when N becomes large there is an optimum amount of iterations where the Pearson coefficient between c_d and c_1 is high and the slope ω reaches unity. As underlined by the figure, at this state of the network the specific heat is still significantly lower than the one obtained with no regularization, corroborating the importance of a Hebbian initialization for reaching the optimal level of accuracy and robustness of the model.

Figure 13. Specific heat of a model inferred from the activity of $N = 40$ real neurons as a function of the inverse temperature β. The circles report the measure at convergence of the Unlearning regularization; the dots report measures when the algorithm is initialized in a Hebbian fashion and early-stopped when $\omega \simeq 1$. Error bars are neglected for clarity purposes. Choice of the parameters: $N = 40$, $M = 10^5$, $\lambda = 10^{-3}$.

7. Conclusions

In this study we have shown that:

- In the context of BMs, and specifically for what concerns the robustness under rescaling of the parameters of the inferred network, the standard L_1 and L_2 regularization techniques are outperformed by the unlearning regularization. This result is valid for both cases where training is stopped earlier, to maximize the similarity between the inferred model and the ground-truth, and at convergence of the algorithms.
- A particular case of the unlearning regularization (i.e. for a vanishing parameter a) reproduces a thermally averaged Hebbian Unlearning rule, that shows good inferential capabilities when initialized in the standard Hebbian fashion. Hebbian Unlearning can be interpreted as a two-steps Boltzmann-Machine learning and it is well reproduced by a L_2 regularization with a high regularization rate. In this case, our study stresses the necessity of early-stopping the training procedure after a specific amount of iterations where the performance reaches its optimum.

The goal of this article was thus to show the effectiveness of unlearning as a form of regularization in BMs. The analysis has been carried out on both small networks, that allowed to compute all quantities without error by exact enumeration, and large networks, generally affected by statistical noise. For the most part of our work the ground-truth distribution was defined by one of two models, either the CW or the SK models, which are known to show a phase transition at a critical temperature. As a result, we could not only test the proximity of the inferred model to the original one, but also the susceptibility of the network under variations of the parameters, by comparing the trend of the specific heat of the ground truth model with respect to the inverse temperature, with the trend obtained after inference with regularization. In addition to this controlled framework, a real-world case, with data sampled from real neuronal activity, has been considered to confirm the results. Both heuristic and empirical arguments support the idea that models inferred via BM learning tend to be pathologically critical [7, 8, 16], i.e. less effective in sampling new examples. On the other hand, the regularized model is much less critical than the one originally learned in [16] from the same data-set. Furthermore, we have displayed that even in the high-dimensional case one can characterize the performance of the algorithm in time by tracking two quantities: the Pearson coefficient and the slope of the empirical covariances versus the thermal correlations. These two observables are fast to compute and allow to establish an early stopping criterion that, combined with a Hebbian initialization of the parameters, reaches a greater generalization and robustness performance. We conclude that, given a high consistency between the model and the ground-truth distribution, the stability of the model under rescaling of the parameters is enhanced by the unlearning regularization. As an alternative strategy to the unlearning regularization presented here, we also evaluated the minimization of the KL distance between the models at $\beta = 1$ and $\beta = a$. Even if the gradient descent equations look similar to the current case, they contain the

contribution of four-point correlations among spins, resulting in a significant increase of the computational cost of the training.

We also analyzed the particular limit of the regularization that gives as learning rules the ones expressed in equations (29) and (30). The resulting procedure strongly resembles the traditional Hebbian Unlearning routine, but with a thermal two-point correlation computed at $\beta = 1$ replacing that computed on the zero-temperature fixed points of the dynamics. Previous work [13] supports the fact that the associative memory performance of such a *thermal* unlearning does not deviate much from the original rule. Unlearning is not new to be mentioned in the statistical inference realm [35–37], yet we are the first to consider the original rule proposed by Hopfield in [9]. As a new contribution to the characterization of unlearning, we tested its inferential power, i.e. the capability of learning the original model. So far, to the best of our knowledge, the original unlearning algorithm was exclusively tested in an associative memory framework, where data are actually patterns, that are sub-dominant in number with respect to the entire set of possible configurations. Our results show a good inferential performance of the algorithm. The explanation for this success relies on the choice of the initialization of the parameters. A Hebbian initialization of the network has two implications. First, the contribution to the gradient of the loss deriving from the data (i.e. the positive *Hebbian* contribution in the BM learning in equation (8)) is much smaller than the contribution deriving from the cross-entropy of the model given the data (i.e. the negative *unlearning* contribution in the BM learning). This condition holds until an optimal amount of iterations that can be estimated from the numerical simulations. Second, we showed that a Hebbian initialization of the couplings points the gradient of the parameters towards the correct direction, i.e. the one of the ground-truth of the inferential problem, at variance with other kinds of parameter initialization. The beneficial effects of a Hebbian initialization for these type of machines has already emerged in the very recent literature [12, 38]. Yet, the topological properties of the Hebbian free energy landscape that make them such a good starting point for training have not received due attention.

We conclude that BM learning can be performed in two steps: an initial overshooting along the *direction of the data* and then a gradual adjustment along the *direction of the model* with increments being updated at each step of the algorithm. The effectiveness of a two-step training for a BM encourages a biological interpretation of this kind of training, in agreement with the hypothesis that synapses in the brain could be fine tuned through the alternation of daily online experience and offline sleep [39, 40]. Such a conjecture is inspiring members of the A.I. community in the development of training procedures for deep neural networks that might substitute the slow and non-biological back-propagation techniques [35, 41]. These observations, together with the importance of the Hebbian initialization of the network, appear to be consistent with previous results in literature concerning the unlearning algorithm and its implementation in associative memory modeling [10–12, 38].

Data availability statement

No new data were created or analyzed in this study.

Acknowledgments

ORCID iD

Francesco Zamponi ⬤ https://orcid.org/0000-0001-9260-1951

References

[1] Hinton G and Sejnowski T J 1999 *Unsupervised Learning: Foundations of Neural Computation* (The MIT Press)
[2] Hinton G 2007 Boltzmann machine *Scholarpedia* **2** 1668
[3] Cocco S, Monasson R and Zamponi F 2022 *From Statistical Physics to Data-Driven Modelling: With Applications to Quantitative Biology* (Oxford University Press)
[4] Jebara T 2012 *Machine Learning: Discriminative and Generative* (Springer)
[5] Younes L 1996 Synchronous Boltzmann machines can be universal approximators *Appl. Math. Lett.* **9** 109–13
[6] Russ W P *et al* 2020 An evolution-based model for designing chorismate mutase enzymes *Science* **369** 440–5
[7] Mora T and Bialek W 2011 Are biological systems poised at criticality? *J. Stat. Phys.* **144** 268–302
[8] Mastromatteo I and Marsili M 2011 On the criticality of inferred models *J. Stat. Mech.* **2011** 10012
[9] Hopfield J J, Feinstein D I and Palmer R G 1983 Unlearning' has a stabilizing effect in collective memories *Nature* **304** 158

[10] Van Hemmen J L, Ioffe L B, Kühn R and Vaas M 1990 Increasing the efficiency of a neural network through unlearning *Physica* A **163** 386

[11] Benedetti M, Ventura E, Marinari E, Ruocco G and Zamponi F 2022 Supervised perceptron learning vs unsupervised Hebbian unlearning: approaching optimal memory retrieval in Hopfield-like networks *J. Chem. Phys.* **156** 104107

[12] Benedetti M and Ventura E 2023 Training neural networks with structured noise improves classification and generalization (arXiv:2302.13417)

[13] Nokura K 1996 Paramagnetic unlearning in neural network models *Phys. Rev.* E **54** 5571

[14] Bishop C M 1995 *Neural Networks for Pattern Recognition* (Clarendon)

[15] Cocco S, Leibler S and Monasson R 2009 Neuronal couplings between retinal ganglion cells inferred by efficient inverse statistical physics methods *Proc. Natl Acad. Sci.* **106** 14058–62

[16] Schneidman E, Berry M, Segev R and Bialek W 2006 Weak pairwise correlations imply strongly correlated network states in a neural population *Nature* **440** 1007–12

[17] Morcos F, Pagnani A, Lunt B, Bertolino A, Marks D S, Sander C, Zecchina R, Onuchic J N, Hwa T and Weigt M 2011 Direct-coupling analysis of residue coevolution captures native contacts across many protein families *Proc. Natl Acad. Sci.* **108** E1293–301

[18] Chen X, Winiarski M, Puścian A, Knapska E, Mora T and Walczak A M 2023 Modelling collective behavior in groups of mice housed under semi-naturalistic conditions (arXiv:2023.07.26.550619)

[19] Sims K 1994 Evolving virtual creatures *Proc. of the 21st Annual Conference on Computer Graphics and Interactive Techniques*

[20] Brock A, Donahue J and Simonyan K 2019 Large scale GAN training for high fidelity natural image synthesis (arXiv:1809.11096)

[21] Kawthekar P 2017 *Evaluating generative models for text generation* PhD Thesis (Standford University)

[22] Sohl-Dickstein J, Weiss E, Maheswaranathan N and Ganguli S 2015 Deep unsupervised learning using nonequilibrium thermodynamics *Proc. 32nd Int. Conf. on Machine Learning* (PMLR) pp 2256–65

[23] Jordan M I and Sejnowski T J 2001 *Graphical Models: Foundations of Neural Computation* (MIT Press)

[24] Barrat-Charlaix P, Muntoni A P, Shimagaki K, Weigt M and Zamponi F 2021 Sparse generative modeling via parameter reduction of Boltzmann machines: application to protein-sequence families *Phys. Rev.* E **104** 024407

[25] Tkačik G, Mora T, Marre O, Amodei D, Palmer S E, Berry M J and Bialek W 2015 Thermodynamics and signatures of criticality in a network of neurons *Proc. Natl Acad. Sci.* **112** 11508–13

[26] Landau L D and Lifshitz E M 1980 *Statistical Physics* (Elsevier Science)

[27] Crick F and Mitchison G 1983 The function of dream sleep *Nature* **304** 111

[28] Van Hemmen J L and Klemmer N 1992 Unlearning and Its Relevance to REM Sleep: Decorrelating Correlated Data *Neural Network Dynamics*, ed J G Taylor, C L T Mannion, J G Taylor, E R Caianiello, R M J Cotterill and J W Clark (Springer) p 30

[29] Wimbauer S, Klemmer N and van Hemmen J L 1994 Universality of unlearning *Neural Netw.* **7** 261

[30] Kleinfeld D and Pendergraft D 1987 Unlearning increases the storage capacity of content addressable memories *Biophys. J.* **51** 47

[31] Peretto P 1984 Collective properties of neural networks: a statistical physics approach *Biol. Cybern.* **50** 51–62

[32] Gardner E 1986 Structure of metastable states in the Hopfield model *J. Phys. A: Math. Gen.* **19** L1047

[33] Amit D J 1989 *Modeling Brain Function* (Cambridge University Press)

[34] Sherrington D and Kirkpatrick S 1975 Solvable model of a spin-glass *Phys. Rev. Lett.* **35** 1792

[35] Hinton G 2022 The forward-forward algorithm: some preliminary investigations (arXiv:2212.13345)

[36] Fachechi A, Barra A, Agliari E and Alemanno F 2022 Outperforming RBM feature-extraction capabilities by 'dreaming' mechanism *IEEE Trans. Neural Netw. Learn. Syst.* **35** 1–10

[37] Agliari M, Aquaro E, Alemanno F and Fachechi A 2023 Regularization, early-stopping and dreaming: a Hopfield-like setup to address generalization and overfitting (arXiv:2308.01421)

[38] Pozas-Kerstjens A, Muñoz-Gil G, Piñol E, García-March M A, Acín A, Lewenstein M and Grzybowski P R 2021 Efficient training of energy-based models via spin-glass control *Mach. Learn.: Sci. Technol.* **2** 025026

[39] Payne J and Nadel L 2019 Sleep, dreams and memory consolidation: the role of the stress hormone cortisol *Learn. Mem.* **11** 671

[40] Girardeau G and Lopes-dos Santos V 2021 Brain neural patterns and the memory function of sleep *Science* **374** 560–4

[41] Scellier B and Bengio Y 2017 Equilibrium propagation: bridging the gap between energy-based models and backpropagation *Front. Comput. Neurosci.* **11** 24