Daydreaming Hopfield Networks and their surprising effectiveness on correlated data

Ludovica Serricchio[1] **Claudio Chilin**[1] **Dario Bocchi**[1] **Raffaele Marino**[2]

Matteo Negri[13 *] **Chiara Cammarota**[1] **Federico Ricci-Tersenghi**[134]

[1] University of Rome 'La Sapienza'
Department of Physics

[2] Università degli Studi di Firenze
Department of Physics

[3] CNR-Nanotec
Rome Unit

[4] INFN
Sezione di Roma

Abstract

In order to improve the storage capacity of the Hopfield model, we develop a version of the dreaming algorithm, called *daydreaming*, that is not destructive and that converges asymptotically to a stationary coupling matrix. When trained on random uncorrelated examples, the model shows optimal performance in terms of the size of the basins of attraction of stored examples and the quality of reconstruction. We also train the *daydreaming* algorithm on correlated data obtained via the random-features model and argue that it exploits the correlations to increase even further the storage capacity and the size of the basins of attraction.

Hopfield Networks [1] are one of the most studied architectures for storing and retrieving patterns in a neural network. The original Hopfield model is based on the Hebb rule [2] which is analytically tractable [3, 4] and whose corresponding dynamics is biologically plausible [5]. Although the number of patterns P that can be stored in the Hopfield model is extensive, i.e. linear in the number of neurons N, its storage capacity $\alpha = P/N \simeq 0.138$ is pretty low [4]. Given this limitation of the Hebb rule, it is crucial to find other learning rules that can improve the storage capacity.

This work elaborates on a storing strategy of the Hopfield model called *dreaming* (or *unlearning*) [6]: the Hebb rule is interpreted as a "day" phase where the desired memories are encoded in the synapses, and the dreaming procedure is interpreted as a "night" phase where spurious memories are erased. This approach was inspired by the hypothesis that, during the REM sleep phase, the human brain erases useless memories while strengthening useful ones [7].

Most versions of this iterative procedure encounter problems repeating the unlearning steps too many times: after a number of iterations that depends on the load α, also desired memories start to get deteriorated and the network faces a catastrophic forgetting, after which no memory can be retrieved anymore [8, 9]. Additionally, the usual dreaming procedure is not very effective in dealing with correlated examples [10].

Inspired by the concept of reinforcing the memories studied in [11], in this work we design a procedure that can be iterated indefinitely, for which no assumption is necessary on the structure and correlation of patterns. The procedure avoids the need for a fine-tuning and only depends on a single

*Corresponding author, matteo.negri@uniroma1.it

Associative Memory & Hopfield Networks in 2023. NeurIPS 2023 workshop.

parameter, while keeping the conceptual simplicity of the original dreaming procedure. We call it *daydreaming*, as it drops any distinction between night and day cycles.

We test daydreaming on uncorrelated data and on the so-called *random-features* (or *hidden-manifold*) correlated data [12]: being the superposition of features, they have been proposed as more realistic model of datasets typically used in machine learning [13]. Moreover, it has been shown in [14] that this data structure enriches the phase diagram of the Hopfield model. For these reasons, this data offers a good preliminar playground to test the retrieval performances of our algorithm, before testing it on real data where less clear baselines are available.

Model, Algorithm and Data

Hopfield networks A Hopfield network is a recurrent neural network made of N neurons $\{s_i\}_{i=1}^N$ that can be in the states ± 1 depending on the signal that they receive from all the other neurons. A each time step k, every neuron is updated with the rule

$$s_i^{(k+1)} = \text{sign}\left(\sum_{j=1}^{N} J_{ij} s_j^{(k)}\right), \tag{1}$$

where J_{ij} is the matrix of synaptic weights. We consider a symmetric matrix ($J_{ij} = J_{ji}$) with zeros on the diagonal ($J_{ii} = 0$). We perform asynchronous updates, meaning that, at each time step, we update the neurons one at at time in random order. The dynamics stops as soon as all the spins reached a fixed point.

This model works as an associative memory if, when initialized to a noisy version of one of P examples $\{\boldsymbol{\xi}^\mu\}_{\mu=1}^P$, the model converges to the clean version of such example. We are interested in finding the synaptic weights that maximize the number of retrievable examples.

Daydreaming algorithm Our algorithm to increase the storage capacity of an Hopfield network consist in "dreaming away" spurious memories and reinforcing good ones at the same time. The removal part operates as in the original dreaming procedure [6]: at each step u, we initialize the network to a random configuration, then we run the update rule in eq. 1 until we reach a fixed point $\boldsymbol{\sigma}^{(u)}$, then we increase its energy. Simultaneously, we reinforce one of the memories, i.e. we choose at random one of the indices μ and we decrease the energy of $\boldsymbol{\xi}^\mu$. In other words, the daydreaming update rule reads

$$J_{ij}^{(u+1)} = J_{ij}^{(u)} + \frac{1}{\tau N}(\xi_i^{\mu(u)}\xi_j^{\mu(u)} - \sigma_i^{(u)}\sigma_j^{(u)}), \tag{2}$$

where τ is a timescale parameter that acts as an inverse learning rate and we divided by N so that the rule scales well with different number of neurons. Moreover, we normalize J_{ij} every N steps.

An uninformed choice for the initialization of J_{ij} would be to sample its elements with a Gaussian distribution, but we found that initializing J_{ij} with the Hebb rule makes the training converge faster without changing the retrieval properties.

The complete pseudo-code is reported in alg. 1.

Random-features data We study examples $\boldsymbol{\xi}^\mu$ generated as a superpositions of D random features $\mathbf{f}^k \in \{-1, +1\}^N$, namely $\xi_i^\mu = \text{sign}(\sum_{k=1}^D c_k^\mu f_i^k)$ where $c_k^\mu \sim \mathcal{N}(0, 1)$ and $f_{ki} \sim \text{Unif}(\{+1, -1\})$.

In this model, in addition to the usual load parameter α, the behaviour is also controlled by the parameter $\alpha_D = D/N$, which describes how strongly correlated the examples are: if $\alpha_D \gg 1$ the distribution of the examples converges to $\text{Unif}(\{+1, -1\})$ and we get back a dataset of uncorrelated examples; while if $\alpha_D \lesssim 1$ the examples are correlated, and the task of storing them is expected to be more difficult. For instance, the storage capacity of the Hebb rule is decreased [14].

Results

To describe the performances of the daydreaming algorithm, we observe the *magnetization* m^μ (or overlap) of a configuration $\boldsymbol{s}$ with a given example $\boldsymbol{\xi}^\mu$, defined as $m^\mu = \frac{1}{N}\sum_{i=1}^N \xi_i^\mu s_i$. We

Algorithm 1 Daydreaming learning algorithm

Require: examples $\{\boldsymbol{\xi}^\mu\}_{\mu=1}^P$
  $J_{ij} \leftarrow \frac{1}{N}\sum_\mu \xi_i^\mu \xi_j^\mu$ ▷ Initialization to the Hebb rule
  $J_{ii} \leftarrow 0$
  for $t = 1, \dots, E$ **do** ▷ Do E epochs
    for $u = 1, \dots, N$ **do** ▷ Do N steps in each epoch
      $\mu \leftarrow \mathrm{Unif}(\{1, \dots, P\})$ ▷ Pick an example at random
      $\sigma_i \leftarrow \mathrm{Unif}(\{+1, -1\})$ ▷ Initialize σ_i at random
      while not converged **do**
        $\sigma_i \leftarrow \mathrm{sign}(\sum_j J_{ij}\sigma_j)$ ▷ Run the dynamics
      end while
      $J_{ij} \leftarrow J_{ij} + \frac{1}{\tau N}(\xi_i^\mu \xi_j^\mu - \sigma_i \sigma_j)$ ▷ Update the coupling matrix
      $J_{ii} \leftarrow 0$
    end for
    $J_{ij} \leftarrow J_{ij}/||J||_2$ ▷ Normalize after each epoch
  end for

initialize the network on a configuration that has initial magnetization m_I with an example, we run the dynamics in eq. 1 until convergence, then we measure the final magnetization m_F with the chosen example. The resulting curves are shown in fig. 1 and are called *retrieval maps*. In particular pure examples are stable if $m_\mathrm{I} = 1$ corresponds to a $m_\mathrm{F} \simeq 1$. Moreover, we ask how much noise we can inject (*i.e.* changing the sign of neurons at random) in an example ($m_\mathrm{I} \in [0, 1]$) before the network stops being able to retrieve it. We therefore define the basin of attraction of a given example as the set of configurations that are mapped to such example by the dynamics. The size of such basins can be deduced by the minimum m_I at which is still possible to retrieve an example, *i.e.* having $m_\mathrm{F} \simeq 1$.

Convergence We show in the appendix (fig. A.1) that daydreaming does not need to be fine-tuned, as once τ is big enough the dynamics of the synaptic matrix does not depend on τ anymore. Additionally, we see that we can iterate indefinitely the update rule in eq. 1 without losing the retrieval capabilities of the network (fig. A.1 and fig. A.2 in the appendix).

Retrieval of uncorrelated data We find that Daydreaming matches state-of-the-art results in retrieving uncorrelated examples. We show with red curves in panels (a), (b) and (c) of fig. 1 the retrieval maps of uncorrelated data for different values of α. In panel (a) we see that, at $\alpha < 0.138$, daydreaming increases the basins of attraction. In panel (b) we see that daydreaming creates large basins of attraction for $\alpha > 0.138$; we match the retrieval map showed in [15]. In panel (c) we show that daydreaming also matches the results described in [11] creating (vanishing) basins of attraction even when the capacity approaches the bound $\alpha = 1$, above which is impossible to store uncorrelated examples [16]. We show retrieval maps for more values of α in fig. A.2 in the appendix.

Retrieval of correlated data We show with blue curves in panels (a), (b) and (c) of fig. 1 the retrieval maps of correlated data for different values of α (see fig. A.3 in the appendix for more values of α). Surprisingly, we find that daydreaming produces basins of attraction of correlated examples that are larger than the ones of uncorrelated examples, at any α we tested. In general, daydreaming extends the storage phase of correlated examples described in [14]. In particular, note from panel (c) and fig. A.3b in the appendix that the retrieval of feature is possible at $\alpha \geq 1$, meaning that the model is somehow exploiting the correlation between examples to overcome the bound at $\alpha = 1$.

Retrieval of features Given that in [14] the authors showed that the Hebb rule is capable of retrieving hidden features for certain values of α and α_D, we ask if the same happens with daydreaming. To do this, we define a *feature magnetization* $\mu_k = \frac{1}{N}\sum_{i=1}^N f_{ki} s_i$ and we study the retrieval maps of the features. We show in panel (d) of fig. 1 that daydreaming produces large basins of attraction for the features, even outside the region in the α, α_D plane described in [14].

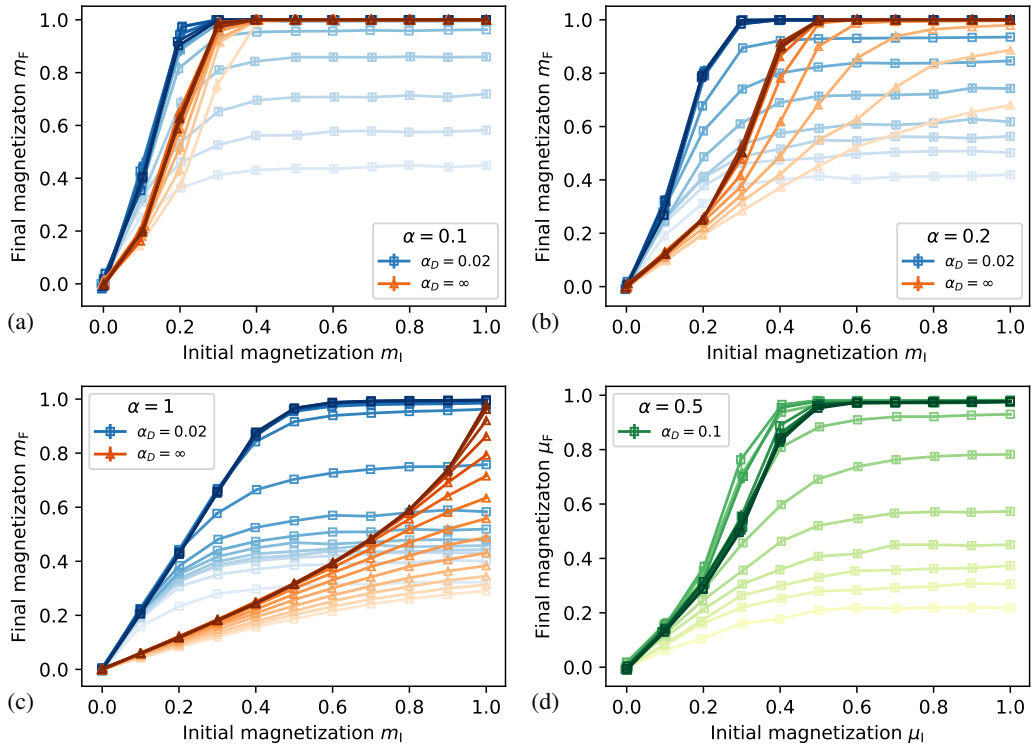

Figure 1: **For the Daydreaming algorithm, correlated examples are easier to store and retrieve than random ones**. We show the retrieval maps for correlated (blue curves) and uncorrelated (red curves) memories during the daydreaming procedure. Different shades of the colors represent different timestamps: the lightest color is $t = 1$ and the darkest is $t = 32768$ (logarithmic spacing). Panel (a) shows results for $\alpha < 0.138$, where daydreaming enlarges the basin of attraction of uncorrelated examples. Panel (b) shows results for $\alpha > 0.138$, where uncorrelated data become stable faster but correlated data end up with a larger basin of attraction. Panel (c) shows results for $\alpha = 1$, where uncorrelated data become stable attractors at the end of the training but their basin of attraction is very small. Panel (d) shows the retrieval map for features hidden in the data. We used $N = 1000$ for these figures.

Conclusions, discussion and perspectives

In order to overcame a series of problems that afflicted the dreaming algorithms used to increase the storage capacity of Hopfield networks, we designed a new learning procedure called daydreaming.

Daydreaming is closely related to the maximum likelihood principle, as its update rule resembles a way to satisfy a moment-matching condition (see for example [17], where the "day" and "night" terms are explicitly identified). In this spirit, some learning rules related to ours (but less effective) have been discussed in [18, 19] for a fully-connected symmetric model and generalized to sparse and asymmetric models in [20, 21]. Moreover, in [22] they discuss an update rule that looks similar to ours, but the way authors sample spurious states is different and leads to smaller basins of attraction. Daydreaming is also related to algorithms of the contrastive divergence family [23], that are commonly used to train Restricted Boltzmann Machines (for some reviews, see for example [24] or [25]).

Daydreaming proved to be a compact, straightforward, and streamlined algorithm, with the convergence rate notably contingent only on the parameter τ. It does not suffer from the problem of dreaming too much, nor requires any fine tuning. Moreover, it seemingly does not require any assumption on the structure of the data, as it finds large basins of attraction even for highly-correlated random-features examples.

This last point is somewhat surprising, as the classical picture in the literature is that correlation hinders retrieval [26, 27, 28, 29, 10]. Another surprising result is that correlated examples have larger

basins of attraction than uncorrelated examples, and can be retrieved above $\alpha = 1$, which is the hard limit for uncorrelated examples [16].

For these reasons, we hypothesize that daydreaming is using some non-trivial mechanism to store correlated examples efficiently, possibly exploiting the features hidden in the data. This is supported by the fact that daydreaming improves the retrieval of features too. Note that this fact is again non-trivial, since the update rule in eq. 1 is ignorant about the internal structure of the examples and only tries to reinforce the examples that is given explicitly.

Given the surprising results of daydreaming on correlated data and its closeness to well-established method to train Boltzmann Machines, it would be interesting to test it on more realistic datasets (such as MNIST or even CIFAR10). In particular, it would be crucial to understand the mechanism in which the network builds basins of attraction and if it indeed exploits hidden features.

Acknowledgements

We thank Enrico Ventura for pointing out a lot of interesting literature. MN acknowledges the support of LazioInnova - Regione Lazio under the program Gruppi di ricerca 2020 - POR FESR Lazio 2014-2020, Project NanoProbe (Application code A0375-2020-36761), and, starting from August 1st 2023, PNRR MUR project PE0000013-FAIR. R.M. is supported by #NEXTGENERATIONEU (NGEU) and funded by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), project MNESYS (PE0000006) "A Multiscale integrated approach to the study of the nervous system in health and disease" (DN. 1553 11.10.2022).

References

[1] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A*, 79(8):2554–8, 1982.

[2] Donald O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, 1949.

[3] Daniel J Amit, Hanoch Gutfreund, and Haim Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55(14):1530, 1985.

[4] Daniel J Amit, Hanoch Gutfreund, and Haim Sompolinsky. Statistical mechanics of neural networks near saturation. *Annals of physics*, 173(1):30–67, 1987.

[5] Daniel J Amit and Daniel J Amit. *Modeling brain function: The world of attractor neural networks*. Cambridge university press, 1989.

[6] J. J. Hopfield, D. I. Feinstein, and R. G. Palmer. "unlearning" has a stabilizing effect in collective memories. *Nature*, 304(5922):158–159, 1983.

[7] Francis Crick and Graeme Mitchison. The function of dream sleep. *Nature*, 304:111–114, 1983.

[8] JL Van Hemmen, LB Ioffe, R Kühn, and M Vaas. Increasing the efficiency of a neural network through unlearning. *Physica A: Statistical Mechanics and its Applications*, 163(1):386–392, 1990.

[9] JL Van Hemmen and N Klemmer. Unlearning and its relevance to rem sleep: Decorrelating correlated data. In *Neural Network Dynamics: Proceedings of the Workshop on Complex Dynamics in Neural Networks, June 17–21 1991 at IIASS, Vietri, Italy*, pages 30–43. Springer, 1992.

[10] JL Van Hemmen. Hebbian learning, its correlation catastrophe, and unlearning. *Network: Computation in Neural Systems*, 8(3):V1, 1997.

[11] Alberto Fachechi, Elena Agliari, and Adriano Barra. Dreaming neural networks: forgetting spurious memories and reinforcing pure ones. *Neural Networks*, 112:24–40, 2019.

[12] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10(4):041044, 2020.

[13] Federica Gerace, Luca Saglietti, Stefano Sarao Mannelli, Andrew Saxe, and Lenka Zdeborová. Probing transfer learning with a model of synthetic correlated datasets. *Machine Learning: Science and Technology*, 3(1):015030, 2022.

[14] Matteo Negri, Clarissa Lauditi, Gabriele Perugini, Carlo Lucibello, and Enrico Malatesta. The hidden-manifold hopfield model and a learning phase transition. *arXiv preprint arXiv:2303.16880*, 2023.

[15] Marco Benedetti, Enrico Ventura, Enzo Marinari, Giancarlo Ruocco, and Francesco Zamponi. Supervised perceptron learning vs unsupervised hebbian unlearning: Approaching optimal memory retrieval in hopfield-like networks. *J Chem Phys*, 156:104107, 2022.

[16] Elizabeth Gardner. The space of interactions in neural network models. *Journal of physics A: Mathematical and general*, 21(1):257, 1988.

[17] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[18] Tetsuya Kojima, Hidetoshi Nonaka, and Tsutomu Da-Te. Capacity of the associative memory using the boltzmann machine learning. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 5, pages 2572–2577. IEEE, 1995.

[19] Carlo Baldassi, Federica Gerace, Luca Saglietti, and Riccardo Zecchina. From inverse problems to learning: a statistical mechanics approach. In *Journal of Physics: Conference Series*, volume 955, page 012001. IOP Publishing, 2018.

[20] Alfredo Braunstein, Abolfazl Ramezanpour, Riccardo Zecchina, and Pan Zhang. Inference and learning in sparse systems with multiple states. *Physical Review E*, 83(5):056114, 2011.

[21] Luca Saglietti, Federica Gerace, Alessandro Ingrosso, Carlo Baldassi, and Riccardo Zecchina. From statistical inference to a differential learning rule for stochastic neural networks. *Interface Focus*, 8(6):20180033, 2018.

[22] G Pöppel and U Krey. Dynamical learning process for recognition of correlated patterns in symmetric spin glass models. *Europhysics Letters*, 4(9):979, 1987.

[23] Miguel Á. Carreira-Perpiñán and Geoffrey Hinton. On contrastive divergence learning. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pages 33–40. PMLR, 06–08 Jan 2005. Reissued by PMLR on 30 March 2021.

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

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[26] Daniel J Amit, Hanoch Gutfreund, and Haim Sompolinsky. Information storage in neural networks with low levels of activity. *Physical Review A*, 35(5):2293, 1987.

[27] José Fernando Fontanari and WK Theumann. On the storage of correlated patterns in hopfield's model. *Journal de Physique*, 51(5):375–386, 1990.

[28] R Der, VS Dotsenko, and B Tirozzi. Modified pseudo-inverse neural networks storing correlated patterns. *Journal of Physics A: Mathematical and General*, 25(10):2843, 1992.

[29] Matthias Löwe. On the storage capacity of hopfield models with correlated patterns. *The Annals of Applied Probability*, 8(4):1216–1250, 1998.

A Appendix: Supplemental figures

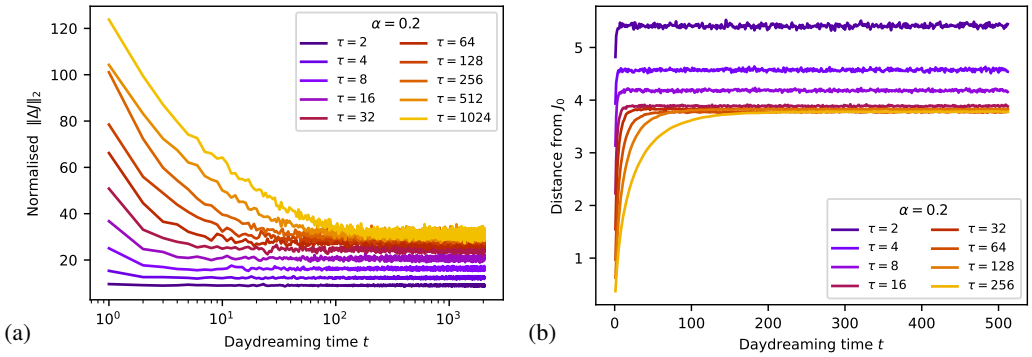

Figure A.1: In panel (a) we show the rescaled norm of the increment of the synaptic matrix $\tau||\Delta J||_2$ as a function of the training time. We see that when $\tau \geq 128$ the dynamics does not depend on τ anymore and the dynamics enters a stationary regime if the training is long enough. In panel (b) we show the distance of the coupling matrix from the Hebbian initial condition J_0, as a function of the training time. Different colors correspond to different values of the characteristic time τ. Note that for $\tau = 64$ the algorithm finds solutions at the same distance as runs with $\tau > 64$, suggesting that, once τ is small enough, decreasing it further only slows down the training.

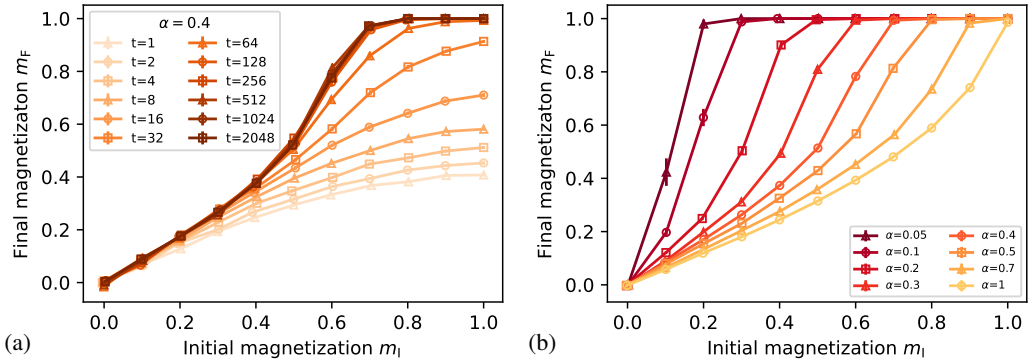

Figure A.2: We show the retrieval maps for uncorrelated examples. Panel (a): we show the evolution of the retrieval map during the training (from lighter to darker shades). The training converges around $t = 128$ and finds basin of attractions that are consistent with the state of the art results in [15]. Panel (b): we show retrieval maps at the end of the training procedure for various values of the load α. Since the convergence time increases with α, each line has a different training time. We used $N = 1000$ for these figures.

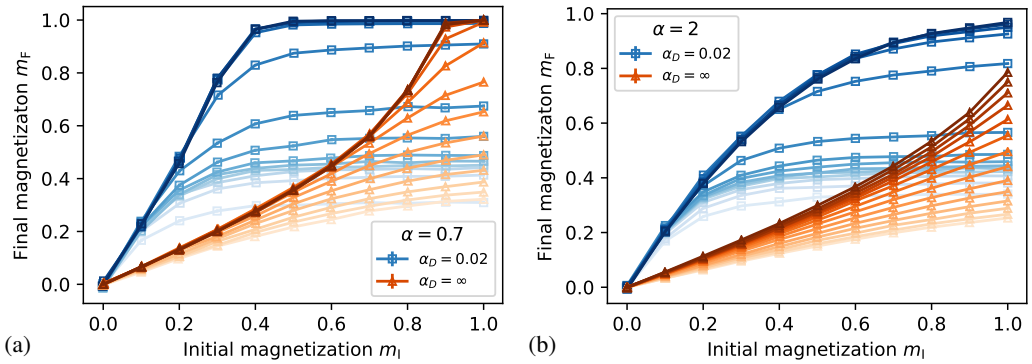

Figure A.3: We show the retrieval maps for correlated (blue curves) and uncorrelated (red curves) memories during the daydreaming procedure. Different shades of the colors represent different timestamps: the lightest color is $t = 1$ and the darkest is $t = 32768$. We used $N = 1000$ for these figures. Panel a) $\alpha = 0.7$; panel b) $\alpha = 2.0$.