PSEUDO-LIKELIHOOD PRODUCES ASSOCIATIVE MEMORIES ABLE TO GENERALIZE, EVEN FOR ASYMMETRIC COUPLINGS

Francesco D'Amico, Saverio Rossi, Luca Maria Del Bono & Matteo Negri
Physics Department
Sapienza University of Rome
Piazzale Aldo Moro 5
Rome, 00185, Italy
{francesco.damico,saverio.rossi,
lucamaria.delbono,matteo.negri}@uniroma1.it

ABSTRACT

The classic inference of energy-based probabilistic models by maximizing the likelihood of the data is limited by the difficulty of estimating the partition function. A common strategy to avoid this problem is to maximize the *pseudo-likelihood* instead, that only requires easily computable normalizations. In this work, we offer the perspective that pseudo-likelihood is actually more than just an approximation of the likelihood in inference problems: we show that, at zero temperature, models trained by maximizing pseudo-likelihood are associative memories. We first show this with uncorrelated binary examples, which get memorized with basins of attraction larger than any other known learning rule of Hopfield models. Then, we test this behavior on progressively more complex datasets, showing that such models capitalize on data structure to produce meaningful attractors, which in some cases correspond precisely to examples from the test set.

1 INTRODUCTION

Associative memories (AM) are models that can learn a number of configurations and output them if the dynamics is initialized close enough to one of those configurations. AM appear in different contexts of modern machine learning, such as the attention mechanism (Ramsauer et al., 2020), generative diffusion (Ambrogioni, 2024) and probabilistic modeling (Schaeffer et al., 2024).

Quantifying overfitting in likelihood-based training is an open problem (Catania et al., 2025): the many methods available relate to the quality of the generated samples and the moments of their distribution (Decelle et al., 2025). One of those methods consist in measuring the propensity to generate samples from the training set (Béreux et al., 2025), which can be interpreted as a regime of AM. Still, to the best of our knowledge, the ground states produced by maximizing the likelihood in energy-based models have no clear theoretical connection with training examples. For instance, in the few situations where theory is available, attractors appear uncorrelated with training examples (Decelle & Furtlehner, 2021; Catania et al., 2025).

As we will see, pseudo-likelihood (Besag, 1974; Nguyen et al., 2017) is more clearly understandable in terms of AM than likelihood, and therefore may help to understand overfitting in probabilistic models. Furtermore, pseudo-likelihood is interesting per se, as it was connected with the attention mechanism (Rende et al., 2024; D'Amico & Negri, 2024) and is also a common technique to efficiently train energy-based generative models (Nguyen et al., 2017).

Contributions In this work we show that optimizing pseudo-likelihood is a way to build an AM. Then we offer the perspective that explaining overfitting of probabilistic models in terms of AMs may also provide an intuition to how generalization works. In fact, it was shown recently that even the most basic Hopfield network is capable of generalization by building attractors in correspondence of previously unseen examples (Kalaj et al., 2024). We support this perspective with numeri-

cal results using the simplest architecture possible, aiming to understand fundamentals behaviors of memorization and generalization that may also appear with more complicated architectures.

2 GENERAL SETTING

To explain the simple design of our recurrent network we start from the basics of the maximum likelihood principle. This step will be useful to fix the notation and to introduce the pseudo-likelihood method, which inspires the architecture and dynamics of our model. In the next section we show why this model has a natural interpretation as an associative memory.

Maximum-likelihood principle Be a dataset $\mathcal{D} = \{\boldsymbol{\xi}^\mu\}_{\mu=1}^P$ of examples $\xi = \{\xi_i\}_{i=1}^N$ that we want to model with a probability distribution $p_J(\mathbf{x}) = \psi_J(\mathbf{x}) / \int d\mathbf{y}\, \psi_J(\mathbf{y})$ over a set of variables $\mathbf{x} = \{x_i\}_{i=1}^N$ dependent on a set of parameters J. The *likelihood* of a data point is $p_J(\boldsymbol{\xi}^\mu)$. One can infer J by maximizing the likelihood of all the data points simultaneously, which results in the minimization of the negative log-likelihood (NLL) loss $\mathcal{L} = -\sum_{\mu=1}^P \log p_J(\boldsymbol{\xi}^\mu)$. This loss is difficult to optimize, because it requires the estimation of the normalization $Z_J = \int d\mathbf{x}\, \psi_J(\mathbf{x})$. One of the most common strategies to deal with this term is the family of algorithms related to the minimization of Contrastive Divergence (Hinton, 2002), which exploit fast out-of-equilibrium stochastic processes instead of a standard Monte Carlo processes, which are inefficient as they reach equilibrium very slowly [1] (Agoritsas et al., 2023).

Pseudo-likelihood Another strategy to avoid the estimation of Z_J entirely is to approximate the joint probability as the product of conditionals $p_J(\mathbf{x}) \simeq \prod_i p_i(x_i|\mathbf{x}_{\setminus i})$, where $\mathbf{x}_{\setminus i} = \{x_j\}_{j(\neq i)}$ is the set of all variables except the i-th variable. The advantage is that now the conditionals can be written as[2] $p_i(x_i|\mathbf{x}_{\setminus i}) = \psi(\mathbf{x})/Z_i(\mathbf{x}_{\setminus i})$, where the normalization $Z_i(\mathbf{x}_{\setminus i}) = \int d\mathbf{y}_i\, \psi(y_i, \mathbf{x}_{\setminus i})$ requires only a single integral and therefore is tractable. The probability of a data point within this ansatz is called *pseudo-likelihood* (Besag, 1974). If we plug this expression in the NLL loss we get a *negative log-pseudo-likelihood* (NLpL) loss

$$\mathcal{L} = -\sum_{\mu=1}^{P}\sum_{i=1}^{N} \log p_i(\xi_i^\mu|\xi^\mu{}_{\setminus i}), \tag{1}$$

which is the quantity that we minimize to train the model in all the experiments described in this work.

Gibbs sampling The core idea of a probabilistic model is the possibility of sampling from the inferred distribution. The standard sampling used in the context of pseudo-likelihood in Gibbs sampling (Geman & Geman, 1984), which consists in updating the variables using the conditional probabilities: in practice, at time t one picks a variable x_i to update, then fixes the values of all the other variables, and then samplesx

$$x_i^{(t+1)} \sim p_i(x_i|\mathbf{x}_{\setminus i}^{(t)}). \tag{2}$$

Rather than using eq. 2 for sampling, in this work we will study the recurrent update as an associative memory and consider its storage and retrieval capabilities. This setting is also known as a stochastic neural network and it has been connected to an online optimization of pseudo-likelihood in Saglietti et al. (2018).

3 ASSOCIATIVE MEMORIES FROM PSEUDO-LIKELIHOOD

Energy-based model It is useful to consider an energy-based parametrization of the probability, namely $\psi_J(\mathbf{x}) = \exp\{-\lambda E(\mathbf{x})\}$, where $E(\mathbf{x})$ is an energy function and λ is an inverse temperature. In this way, sampling from $p_J(\mathbf{x})$ can be seen as a stochastic process at thermodynamic equilibrium, and the most probable states can be related to the minima of $E(\mathbf{x})$ using statistical mechanics. Inferring the coupling of $E(\mathbf{x})$ is ofter referred to Boltzmann learning (Ackley et al., 1985), or energy-based probabilistic modeling (LeCun et al., 2006; Song & Kingma, 2021).

[1] Recent advances (Béreux et al., 2025) allow to perform the equilibrium process much faster.

[2] To lighten the notation, we omit the dependence on J of $\psi(\mathbf{x})$, $p_i(x_i|\mathbf{x}_{\setminus i})$ and $Z_i(\mathbf{x}_{\setminus i})$.

Intuition: associative memory optimizes pseudo-likelihood when $\lambda \to \infty$ Energy is still a useful concept when combined with pseudo-likelihood, even if we study retrieval rather than sampling: it provides an intuition that the dynamics in eq. 2 is an associative memory. In fact, if we choose a simple two-body interaction $E(\mathbf{x}) = -\sum_{i \neq j} J_{ij} x_i x_j$ between binary variables $x_i = \pm 1$, the NLpL loss in eq. 1 becomes

$$\mathcal{L} = -\sum_{\mu=1}^{P} \sum_{i} \left[\xi_i^\mu \sum_{j(\neq i)} J_{ij} \xi_j^\mu - \frac{1}{\lambda} \log 2 \cosh \left\{ \lambda \sum_{j(\neq i)} J_{ij} \xi_j^\mu \right\} \right]. \tag{3}$$

The extremization of eq. 3 leads to a set of conditions known as the Callen identities (Callen, 1963) (see Nguyen et al. (2017) for a longer discussion):

$$0 = \frac{\partial \mathcal{L}}{\partial J_{kl}} = \sum_{\mu=1}^{P} \left[\xi_k^\mu - \tanh \left\{ \lambda \sum_{j(\neq k)} J_{kj} \xi_j^\mu \right\} \right] \xi_l^\mu, \tag{4}$$

Note that the couplings inferred with this method are in general not symmetric, and therefore they do not produce an energy function when inserted back in $E(\mathbf{x})$. Since energy is a desirable concept, strategies to symmetrize the couplings are commonly used in this kind of inference (Ekeberg et al., 2013; 2014; Nguyen et al., 2017). In the following, instead, we keep the couplings asymmetric, and we will see that they still produce associative memories. Eq. 4 can be satisfied in the limit $\lambda \to \infty$ by zeroing each term in the square brackets, namely if

$$\xi_k^\mu = \operatorname{sgn} \left\{ \sum_{j(\neq k)} J_{kj} \xi_j^\mu \right\} \quad \forall \mu, k. \tag{5}$$

Eq. 5 is the stability condition for an associative memory with P attractors (it can also be seen as an autoregressive condition, or a self-consistency equation). To fix the ideas: in the case of uncorrelated examples, eq. 5 is satisfied by the Hebb rule $J_{kl} = \frac{1}{\sqrt{N}} \sum_{\mu=1} \xi_k^\mu \xi_l^\mu$ when $P < \alpha_{\rm c} N$, where $\alpha_{\rm c} \simeq 0.14$. Another example solution of eq. 5 is the pseudo-inverse rule (Kanter & Sompolinsky, 1987), valid up to $\alpha_c = 1$. Eq. 5 is not the most general solution eq. 4, therefore it is interesting to test numerically if optimizing NLpL actually produces associative memories at all. Moreover, the Hebb rule is not the most general solution of eq. 5, and a real optimization of loss NLpL may find better basins of attraction and/or higher storage capacity for uncorrelated data. Optimizing NLpL may also produce interesting associative memories when used with correlated data, since associative memories have been recently shown to be capable of generalization. To study these properties, we design the setting described in the next section.

Training at $\lambda = 1$ and running at $\lambda \to \infty$ We train the recurrent network defined in Eq. 2 for binary variables and energy $E(\mathbf{x}) = -\sum_{i \neq j} J_{ij} x_i x_j$, by minimizing loss in Eq. 3 with gradient descent. The value of λ during training has no effect besides rescaling the rows of J. Instead of fixing the norm of J, we set $\lambda = 1$ and we let the coupling evolve freely. Additionally, when we run the dynamics, we set $\lambda \to \infty$, so that the norm of J is irrelevant. We stop the training when the size of the basins of attraction of the training examples do not change anymore (the gradient keeps increasing the norm of J even after the size of the basins converge). Note that, if J were symmetric, the synchronous dynamics with $\lambda \to \infty$ would optimize the energy function described in (Peretto, 1984). In our case, instead, each variable x_i independently aligns to the corresponding so-called local field $h_i(\mathbf{x}_{\setminus i}) = \sum_{j(\neq k)} J_{kj} \xi_j^\mu$, which can be interpreted as the optimization of a *local* energy term. The advantage of having independent updates for each variables is that each row of J can be trained in parallel, giving a substantial numerical advantage.

4 RESULTS

In this section we train the model as described in the previous section on multiple dataset. On those datasets we perform the same experiment: we test the basin of attraction of a chosen configuration $\mathbf{x}^*$ (typically train or test examples). To do this, we initialize the dynamics to a configuration $\mathbf{x}^{\rm IN}$ that has overlap $m_{\rm IN} = \mathbf{x}^{\rm IN} \cdot \mathbf{x}^* / N$ with the chosen configuration. Then we updated it until we reach a fixed point $\mathbf{x}^{\rm F}$. Finally, we measure the overlap $m_{\rm F} = \mathbf{x}^{\rm F} \cdot \mathbf{x}^* / N$. A plot of $m_{\rm IN}$ vs $m_{\rm F}$ is

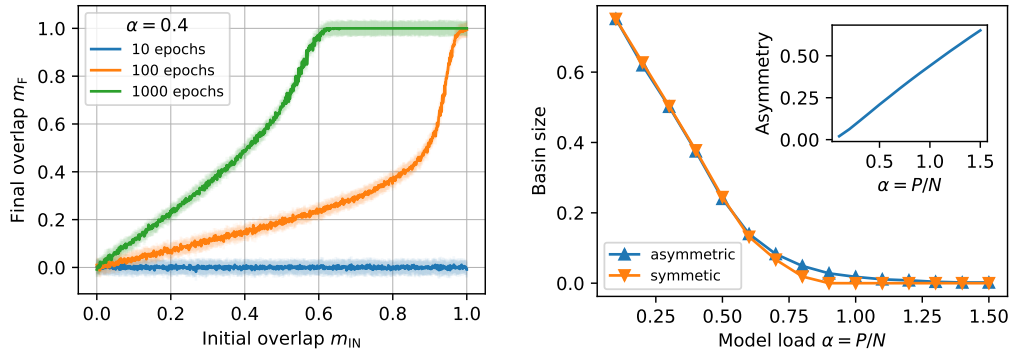

Figure 1: **Pseudo-likelihood produces large basins of attraction of uncorrelated examples.** *Left*: we plot the retrieval map of random i.d.d. examples during the training process for $\alpha = 0.4$. The error bars correspond to the average over the whole dataset for a single instance of training. *Right*: we compare the size of the basins of attraction of asymmetric and symmetric couplings, for various α. The size of the basin is computed as the point where the retrieval map at 1000 epochs goes below $m_F = 0.99$. *Inset*: we plot the asymmetry of the coupling matrix $\|J - J^\top\|_2/\|J\|_2$ as function of α. In all panels $N = 1000$.

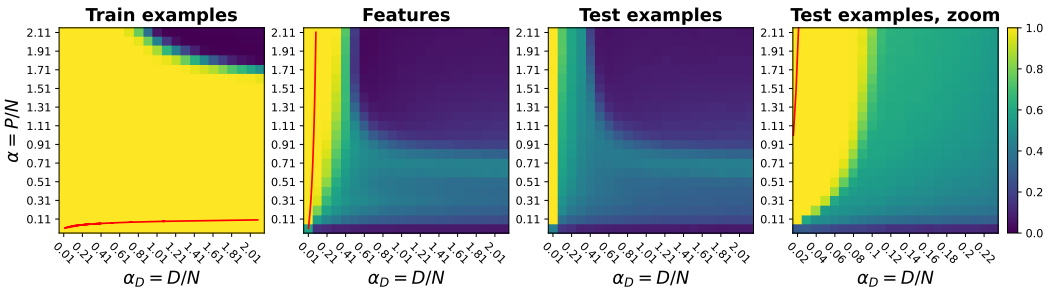

Figure 2: **Pseudo-likelihood improves generalization for correlated random-features examples.** The colors correspond to values m_F given $m_{IN} = 1$, for $\xi_{train}, f, \xi_{test}$, corresponding to storage, learning and generalization phases. Pseudo-likelihood enhances the area of all three phases with respect to Hebb rule, shown as red curves taken from Kalaj et al. (2024). $N = 1000$ in the fist three panels and $N = 2000$ in the last one. $L = 3$ in all panels.

called *retrieval map*. Our goal is to show two different phases, as highlighted in (Kalaj et al., 2024): a *storage phase*, where only training examples are fixed point of the dynamics, and a *generalization phase*, where attractors appear near previously unseen examples. We first show results on synthetic datasets where theoretical thresholds are known (Amit et al., 1987; Kalaj et al., 2024), then we move to two real datasets to check if the same phenomenology is present.

Uncorrelated synthetic examples We train the model with random i.i.d. examples, namely $\xi_i^\mu = \pm 1$ with uniform probability. In this case there cannot be any generalization, as data are uncorrelated, and therefore we only consider storage properties. We can see from fig. 1 that the training procedure produces large basins of attraction around training examples well above the capacity of an Hopfield model (Hopfield, 1982). The size of the basins rapidly approaches zero for $\alpha > 1$ (the theoretical bound for asymmetric couplings is $\alpha = 2$ (Gardner, 1988)). Notably, these basins are equal or larger than those for symmetric couplings.

Correlated synthetic examples We train the model with binary examples ξ_i^μ generated as superposition of binary random features $f_{ki} = \pm 1$ i.i.d. with uniform probability, namely $\xi_i^\mu = \text{sgn}(\sum_{k=1}^D c_k^\mu f_{ki})$. For fixed μ, the coefficients c_k^μ have L non-zero entries, in random locations and

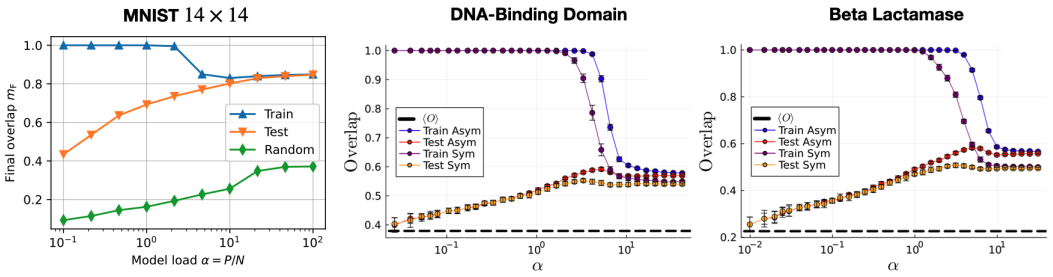

Figure 3: **Pseudo-likelihood is able to generalize even for real data.** We show final overlap of dynamics starting at overlap 1 with training or test examples, as function of dataset size $P = \alpha N$. *Left*: binarized 14×14 MNIST dataset. We also show the curve for images that are random sets of pixels, to check when the network stabilizes every configuration. Error bars are negligible. *Center and Right*: protein sequences from the families DNA-Binding Domain and the Beta Lactamase, respectively. Blue and red correspond to a model with asymmetric couplings, purple and yellow with symmetric. For both, we show train and test examples. The black dashed line is the average overlap between natural sequences. Results are averaged over 20 choices of the training set for each value of α.

uniformly sampled in ± 1, so that each example is the superposition of L features. We see from fig. 2 that pseudo-likelihood greatly surpasses the storage and the generalization phases of the Hopfield model (described in Kalaj et al. (2024)), and even the region where features themselves are stable (described in Negri et al. (2023)). Notice from the fist panel of fig. 2 that, due to data being correlated, it is possible to store them above the theoretical threshold for uncorrelated examples $\alpha = 2$ (Gardner, 1988).

Results on MNIST As a simple instance of real dataset, we train the model using the binarized 14×14 MNIST described in Belyaev & Velichko (2020). In fig. 3 we show that pseudo-likelihood stores training images for a small value of $\alpha = P/N$ (storage phase), P being the size of training dataset. For bigger values of α train and test images produce a final overlap $m_{\mathrm{F}} \simeq 0.85$. By inspecting visually retrieved examples (see fig. 4 in appendix 6), we can see that this value of the final overlap corresponds to a very good memorization.

Results on protein sequences As harder instances of real datasets we study two protein families, the DNA-Binding Domain (DBD) and the Beta Lactamase, as model of protein sequences need to model high-order correlations to generalize well (Trinquier et al., 2021; Decelle et al., 2025). For these datasets, we use an existing library called *plmDCA* (Ekeberg et al., 2013; 2014), see appendix B for the specific details of protein sequence data. The results are plotted in Fig. 3. Similarly to what observed in the other cases, we see that for small values of α the model does not move from its initial state when starting from a sequence in the training set. This memorization region holds up to α between 1 and 10. For larger values, the training sequences are not anymore fixed points of the dynamics and the overlap quickly saturates to the large-load value. Starting from the test sequences, the overlap follows an opposite trend: for small values of α, m_{F} is close to the average overlap between natural sequences, while as the load increases it increases as well. For large values of α, m_{F} saturates to the same point at the one obtained starting from the training sequences. Differently from the previous examples, with proteins we observe m_{F} around 0.55 and 0.6 in the generalization phase, meaning that the attractors of the model are correlated with train and test examples even if the model is unable to retrieve them (which was expected, as these datasets should be much harder).

5 Conclusions

We showed that optimizing pseudo-likelihood can be interpreted as building an associative memory. This connection was already suggested in a neuro-biological setting in Saglietti et al. (2018), where the authors described a moment-matching training procedure as an online optimization of pseudo-likelihood. In this work, instead, we focused on the exact optimization of pseudo-likelihood, showing that the associative it builds is powerful enough to produce attractors in correspondence of

previously unseen examples for simple datasets. For harder datasets, the same model still presents non-trivial attractors, which are as close to training examples as they are to test examples, which still suggests a form of generalization. The fact that an associative memory can be obtained by optimizing a loss function was already discussed in Alemanno et al. (2023), but in that case the authors defined a loss specific for that purpose (which also requires symmetric couplings). Here, instead, we showed that an associative memory (capable of generalization) shows up as an unintended side-effect of a probabilistic model. Overall, we propose that this scheme may be relevant to generalization in modern self-supervised problems such as generative diffusion (Ambrogioni, 2024) and attention mechanism Rende et al. (2024); D'Amico & Negri (2024). Moreover, if a similar perspective could be adopted for maximum-likelihood, it would contribute to understand overfitting and generalization also in Boltzmann machines, helping with the open problems described in Catania et al. (2025).

6 ACKNOWLEDGMENTS

We thank Federico Ricci-Tersenghi for providing the code used to run the parallel tempering simulations. We also thank Dario Bocchi for providing some preliminary data used for comparison. We thank Francesco Zamponi for useful comments. LMDB acknowledges the support of "Bando Ricerca Scientifica 2024 - Avvio alla Ricerca" of Sapienza University. SR acknowledges the support of FIS (Italian Science Fund) 2021 funding scheme (FIS783 - SMaC - Statistical Mechanics and Complexity) from MUR, Italian Ministry of University and Research and from the PRIN funding scheme (2022LMHTET - Complexity, disorder and fluctuations: spin glass physics and beyond) from MUR, Italian Ministry of University and Research. MN acknowledges the support of PNRR MUR project PE0000013-FAIR.

REFERENCES

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Elisabeth Agoritsas, Giovanni Catania, Aurélien Decelle, and Beatriz Seoane. Explaining the effects of non-convergent MCMC in the training of Energy-Based Models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 322–336. PMLR, July 2023. URL https://proceedings.mlr.press/v202/agoritsas23a.html. ISSN: 2640-3498.

Francesco Alemanno, Miriam Aquaro, Ido Kanter, Adriano Barra, and Elena Agliari. Supervised Hebbian learning. *Europhysics Letters*, 141(1):11001, January 2023. ISSN 0295-5075. doi: 10.1209/0295-5075/aca55f. URL https://dx.doi.org/10.1209/0295-5075/aca55f. Publisher: EDP Sciences, IOP Publishing and Società Italiana di Fisica.

Luca Ambrogioni. In search of dispersed memories: Generative diffusion models are associative memory networks. *Entropy*, 26(5):381, 2024.

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

MA Belyaev and AA Velichko. Classification of handwritten digits using the hopfield network. In *IOP conference series: materials science and engineering*, volume 862, pp. 052048. IOP Publishing, 2020.

Nicolas Béreux, Aurélien Decelle, Cyril Furtlehner, Lorenzo Rosset, and Beatriz Seoane. Fast training and sampling of restricted boltzmann machines. In *13th International Conference on Learning Representations-ICLR 2025*, 2025.

Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225, 1974.

H Bo Callen. A note on green functions and the ising model. *Phys. Letters*, 4, 1963.

Giovanni Catania, Aurélien Decelle, Cyril Furtlehner, and Beatriz Seoane. A theoretical framework for overfitting in energy-based modeling. *arXiv preprint arXiv:2501.19158*, 2025.

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

Aurélien Decelle, Alfonso de Jesús Navas Gómez, and Beatriz Seoane. Inferring high-order couplings with neural networks. *arXiv preprint arXiv:2501.06108*, 2025.

Francesco D'Amico and Matteo Negri. Self-attention as an attractor network: transient memories without backpropagation. In *2024 IEEE Workshop on Complexity in Engineering (COMPENG)*, pp. 1–6. IEEE, 2024.

Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer potts models. *Phys. Rev. E*, 87:012707, Jan 2013. doi: 10.1103/PhysRevE.87.012707. URL https://link.aps.org/doi/10.1103/PhysRevE.87.012707.

Magnus Ekeberg, Tuomo Hartonen, and Erik Aurell. Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *Journal of Computational Physics*, 276:341–356, 2014. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2014.07.024. URL https://www.sciencedirect.com/science/article/pii/S0021999114005178.

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

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6): 721–741, 1984.

Geoffrey E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800, August 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018. URL https://ieeexplore.ieee.org/abstract/document/6789337. Conference Name: Neural Computation.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Silvio Kalaj, Clarissa Lauditi, Gabriele Perugini, Carlo Lucibello, Enrico M Malatesta, and Matteo Negri. Random features hopfield networks generalize retrieval to previously unseen examples. *arXiv preprint arXiv:2407.05658*, 2024.

I. Kanter and H. Sompolinsky. Associative recall of memory without errors. *Physical Review A*, 35(1):380–392, January 1987. doi: 10.1103/PhysRevA.35.380. URL https://link.aps.org/doi/10.1103/PhysRevA.35.380. Publisher: American Physical Society.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Matteo Negri, Clarissa Lauditi, Gabriele Perugini, Carlo Lucibello, and Enrico Malatesta. Storage and learning phase transitions in the random-features hopfield model. *Physical Review Letters*, 131(25):257301, 2023.

H. Chau Nguyen, Riccardo Zecchina, and Johannes Berg. Inverse statistical problems: from the inverse Ising problem to data science. *Advances in Physics*, 66(3):197–261, July 2017. ISSN 0001-8732. doi: 10.1080/00018732.2017.1341604. URL https://www.tandfonline.com/doi/full/10.1080/00018732.2017.1341604. Publisher: Taylor & Francis.

P. Peretto. Collective properties of neural networks: A statistical physics approach. *Biological Cybernetics*, 50(1):51–62, February 1984. ISSN 1432-0770. doi: 10.1007/BF00317939. URL https://doi.org/10.1007/BF00317939.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Riccardo Rende, Federica Gerace, Alessandro Laio, and Sebastian Goldt. Mapping of attention mechanisms to a generalized potts model. *Physical Review Research*, 6(2):023057, 2024.

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, October 2018. doi: 10.1098/rsfs.2018.0033. URL https://royalsocietypublishing.org/doi/full/10.1098/rsfs.2018.0033. Publisher: Royal Society.

Rylan Schaeffer, Nika Zahedi, Mikail Khona, Dhruv Pai, Sang Truong, Yilun Du, Mitchell Ostrow, Sarthak Chandra, Andres Carranza, Ila Rani Fiete, et al. Bridging associative memory and probabilistic modeling. *arXiv preprint arXiv:2402.10202*, 2024.

Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Jeanne Trinquier, Guido Uguzzoni, Andrea Pagnani, Francesco Zamponi, and Martin Weigt. Efficient generative modeling of protein sequences using simple autoregressive models. *Nature communications*, 12(1):5800, 2021.

Martin Weigt, Robert A. White, Hendrik Szurmant, James A. Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009. doi: 10.1073/pnas.0805923106. URL https://www.pnas.org/doi/abs/10.1073/pnas.0805923106.

APPENDIX

A RETRIEVAL EXAMPLES FOR MNIST

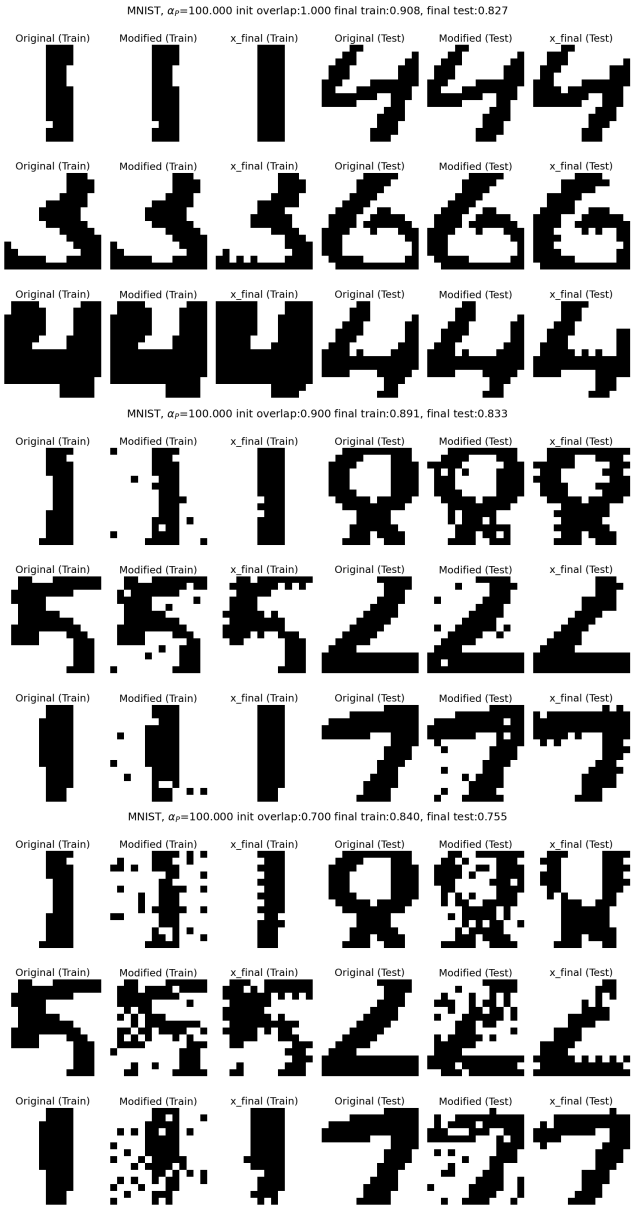

Figure 4: Examples of retrieval of test and train MNIST images with various initial overlaps.

B DETAILS SPECIFIC TO THE DATASETS OF PROTEIN SEQUENCES

We test the procedure described for the other datasets also on data coming from protein sequences. Notice that, in this case, we are not generating synthetic data anymore, instead we are taking as train and test sets some sequences of amino acids that are found in nature. From each protein family we produce a Multiple Sequence Alignment (MSA) in order to have all the sequences with the same length L and aligned between them. In order to account for the bias due to the presence of many similar sequences in the natural MSA, we assign a weight $w_s = 1/n_s$ to each sequence s, with

n_s the number of different sequences in the alignment closer than 80% in length to s. The value $M_{\text{eff}} = \sum_s w_s$ is the effective number of sequences in the alignment. This is a standard procedure when dealing with this kind of data. From this MSA we select a set of P sequences that we use to train the model, drawn from the full MSA with a probability proportional to their weight.

We study in particular two protein families, the DNA-Binding Domain (DBD) and the Beta Lactamase one. The former has a length $L = 76$, a number of sequences $M = 13310$ ($M_{\text{eff}} = 3153$) while for the latter $L = 202$, $M = 18334$ ($M_{\text{eff}} = 6875$).

In the spirit of Direct Coupling Analysis (DCA) (Weigt et al., 2009) each sequence in the MSA is assumed to be drawn from the probability distribution of an equilibrium Potts model with 21 states (corresponding to the 20 amino acids plus the gap symbol '-'). As the direct maximization of the likelihood of such a model is often prohibitive, one resorts to the pseudo-likelihood approximation. The parameters of the model are inferred via the pseudo-likelihood maximization algorithm (plmDCA) described in Ekeberg et al. (2013; 2014).

Once the parameters of the model have been obtained, we proceed by simulating the evolution. Similarly to what we discussed for the other systems, the evolution starts from a certain sequence S^{i} of amino acids and evolves following a 0-temperature dynamics, in which only mutations which increase the probability are accepted. As the dynamics takes place at $T = 0$, the evolution stops after some steps as the system finds itself in a stable state corresponding to a sequence S^{f}. One can then compute the overlap $q(S^{\text{i}}, S^{\text{f}}) = 1/L \sum_{n=1}^{L} \delta_{S^{\text{i}}_n, S^{\text{f}}_n}$ between the final state and the initial one, with $\delta_{a,b}$ the Kronecker delta function.

We study the model obtained from the inference procedure at different values of the load in the range $1 < P < M_{\text{eff}}$. For each value of P, we choose $N_{\text{s}} = 2000$ starting sequences drawn uniformly with repetition from the training set and we let them evolve, comparing the resulting stable states with the starting condition. We then repeat the same steps drawing this time the N_{s} initial sequences from the full MSA, excluding the ones chosen for the training.