Dynamical learning in deep asymmetric recurrent neural networks

Davide Badalotti, Carlo Baldassi, Marc Mézard, Mattia Scardecchia, and Riccardo Zecchina*
Department of Computing Sciences, Bocconi University, Milan, Italy

(Received 9 September 2025; revised 29 December 2025; accepted 27 February 2026; published 11 June 2026)

We investigate recurrent neural networks with asymmetric interactions and demonstrate that the inclusion of self-couplings or sparse excitatory intermodule connections leads to the emergence of a densely connected manifold of dynamically accessible stable configurations. This representation manifold is exponentially large in system size and is reachable through simple local dynamics, despite constituting a subdominant subset of the global configuration space. We further show that learning can be implemented directly on this structure via a fully local, gradient-free mechanism that selectively stabilizes a single task-relevant network configuration. Unlike error-driven or contrastive learning schemes, this approach does not require explicit comparisons between network states obtained with and without output supervision. Instead, transient supervisory signals bias the dynamics toward the representation manifold, after which local plasticity consolidates the attained configuration, effectively shaping the latent representation space. Numerical evaluations on standard image classification benchmarks indicate performance comparable to that of multilayer perceptrons trained using backpropagation. More generally, these results suggest that the dynamical accessibility of fixed points and the stabilization of internal network dynamics constitute viable alternative principles for learning in recurrent systems, with conceptual links to statistical physics and potential implications for biologically motivated and neuromorphic computing architectures.

DOI: 10.1103/lbr6-91sn

I. INTRODUCTION

The remarkable success of contemporary artificial intelligence systems, driven by large-scale artificial neural networks (ANNs), has stimulated intense research into their underlying computational principles. Currently, the dominant training paradigm for ANNs is empirical risk minimization via backpropagation [1]. While effective, this approach requires massive datasets and computational resources, and is generally considered incompatible with the goal of explaining computation in biological neural circuits [2,3]. Furthermore, the architectures underlying today's state-of-the-art models [1,4,5] fail to leverage the rich dynamical behaviors inherent to biological neural networks. A major scientific challenge thus lies in the design of ANNs that learn and operate more like biological brains, without relying on global gradient computation.

To this end, with feedforward architectures, some approaches modify the backward pass by using learned inverses [6] or random projections [7] to propagate the error signal. Others replace it, either with a dynamics that updates activations and weights to make the forward pass internally consistent while forcing a desired output [8,9] or with an additional forward pass on synthetic data that must be distinguished from real data through a learned local criterion [10]. With recurrent architectures, which is the setting of the present work, reservoir computing [11] sidesteps the credit assignment problem by leveraging a fixed recurrent module, while spiking neural networks propose more realistic neuron models but are typically trained via backpropagation through time [12]. The recently proposed neural Langevin machine [13] stabilizes the chaotic dynamics by introducing an Onsager-type feedback term and applies a local contrastive asymmetric learning rule to train a generative model. Finally, equilibrium propagation perturbs the dynamics of an energy-based model and uses a contrastive Hebbian rule to estimate the gradient of a global loss function [14].

In this work, we introduce an optimization-free learning paradigm based on stabilizing fixed points of the dynamics in asymmetric recurrent neural networks. Learning is fully distributed, relies only on local information, and does not require symmetry or a distinction between training and inference. Crucially, the learning step operates on a single task-relevant configuration: Unlike contrastive schemes, it does not compare states obtained with and without output supervision, but directly stabilizes the reached configuration to shape the latent space. Using standard benchmarks, we show that this approach achieves performance comparable to a multilayer perceptron trained with backpropagation with the same number of parameters, highlighting its relevance for artificial intelligence.

Our model is based on a *core module* that is a network of binary neurons interacting with asymmetric couplings. While these models have been studied for a long time in the context of computational neuroscience, most studies have focused on their chaotic regime [15,16], or on the edge of chaos, for signal processing applications. At odds with these approaches,

*Contact author: riccardo.zecchina@unibocconi.it

we introduce an excitatory self-coupling in the core module, or alternatively sparse and strong excitatory couplings between several core modules, and we show that with well-tuned strength of the excitation this leads to the appearance of an accessible, dense, connected cluster of stable fixed points—a high local entropy region [17]—which we call the internal representation manifold (RM).

For simplicity, we adopt a discrete-time, binary formulation. Our results concern the structure of stationary states, for which the discrete-continuous distinction is not essential, and preliminary simulations indicate that the framework extends to continuous time. Binary activations allow us to isolate robust qualitative effects without additional technical complications.

The geometry of the RM can be exploited for learning by mapping input-output associations onto fixed points of the dynamics, as follows. Given an input and a desired output (label), the network starts in a neutral configuration and initially evolves under the influence of the external input. After a short time, a transient supervisory signal from the output appears, driving the dynamics into a fixed point s'. Then, the supervisory signal is removed and the network settles into a second configuration s^*. Finally, learning happens through a local synaptic plasticity rule that stabilizes s^*. As it turns out, in the RM phase, s' and s^* are close to each other and the synaptic reinforcement leads to a stabilization of the input-output association: Exposed to the same input again, without any supervision, the network state will evolve into a configuration that is mapped into the correct output by a learned linear readout. This procedure implements a simple, fully local learning mechanism grounded in the dynamics of asymmetric deep recurrent networks, which does not use error gradient and does not require any additional feedback circuit.

The paper is organized as follows. First, we define the model and present the statistical mechanics analysis concerning the onset of the RM. Next, we define the learning process and present the results of extensive numerical experiments. Finally, we study the role of the RM geometry for learning.

II. CORE MODULE

The core module is a network of binary neurons with state $s \in \{-1, 1\}^N$ and an $N \times N$ diluted asymmetric interaction matrix J. The off-diagonal matrix elements J_{ij}, $i \neq j$ are zero with probability $1 - \rho$, and with probability ρ they are drawn independently from an unbiased Gaussian distribution of variance $\frac{1}{\rho N}$. We introduce a self-interaction term $J_{ii} = J_D \geqslant 0$ for all i, which softens the fixed-point constraints and affects the dynamics. The network evolves in discrete time by aligning each neuron to its local field through the update rule:

$$s_i \leftarrow \operatorname{sgn}\left(\sum_{j \neq i} J_{ij} s_j + J_D s_i\right), \tag{1}$$

with sgn being the sign function. Throughout this study, we adopt a synchronous dynamical scheme for the updates. Due to the asymmetry of the interactions, the dynamics is not governed by a Lyapunov function and the structure of fixed points is subtle. While for $J_D = 0$ the network is chaotic and no fixed points exist, for $J_D > 0$ there exist exponentially many fixed points, which, however, remain unreachable to the dynamics. Above a critical value of J_D, there appears a subdominant cluster of fixed points, the RM, which can be accessed by various types of algorithms including (for large enough J_D) the simple iterative dynamics (1). This phenomenology is reminiscent of what was found in supervised learning with nonconvex, feedforward classifiers [17–20].

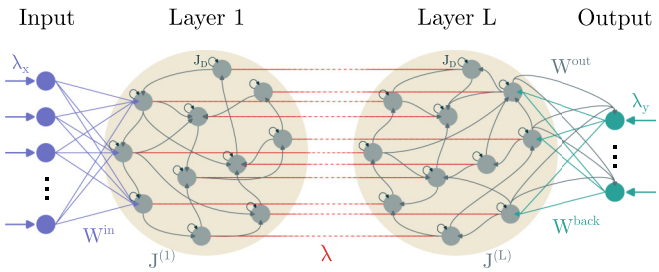

FIG. 1. A multilayer chain model with sparse intermodule excitatory couplings, and input/output layers.

When using such a network for learning, the influence of the input—and possibly of the output label—on the network is mediated by linear input and output layers, respectively. These influence the dynamics of the system and can be trained jointly with the internal couplings using the same local learning rule.

III. MULTILAYER CHAIN

As a minimal model of hierarchical processing, we consider a chain of L core modules like the one above, with states $s^{(l)} \in \{-1, 1\}^N$, and independent interaction matrices $J^{(l)} \in \mathbb{R}^{N \times N}$, $l = 1, \ldots, L$. We introduce a positive interaction between homologous neurons in adjacent layers with uniform strength $\lambda \geqslant 0$:

$$s_i^{(l)} \leftarrow \operatorname{sgn}\left(\sum_{j \neq i} J_{ij}^{(l)} s_j^{(l)} + J_D s_i^{(l)} + \lambda\left(s_i^{(l-1)} + s_i^{(l+1)}\right)\right), \tag{2}$$

with the understanding that $s_i^{(0)} \equiv s_i^{(L+1)} \equiv 0$. This reduces to the core model for $L = 1$. As we shall see, the excitatory sparse interactions across modules governed by λ play a very similar role to the diagonal coupling J_D; in particular, when λ is large enough, the dynamics converges to the RM even when $J_D = 0$. Figure 1 provides a sketch of the chain.

IV. ANALYSIS OF THE GEOMETRY OF FIXED POINTS

To analyze the geometric structure of the fixed points and their accessibility, we first compute the typical number $\mathcal{N}_{\text{fp}}$ of fixed points of the multilayer chain in absence of external inputs, before learning takes place. Since this number is exponentially large, we shall compute the associated entropy density $S_{\text{fp}} = \lim_{N \to \infty} \frac{1}{LN} \ln \mathcal{N}_{\text{fp}}$. As S_{fp} is expected to self-average, we are interested in the quenched average $\frac{1}{LN} \mathbb{E} \ln \mathcal{N}_{\text{fp}}$, where the expectation is taken with respect to the distribution of off-diagonal couplings J_{ij}. It turns out that in the large N limit the "annealed average," i.e., the easily computed first-moment upper bound $\frac{1}{LN} \ln \mathbb{E}\, \mathcal{N}_{\text{fp}}$, is tight for the core model and for $L = 2$. For instance, for the two-layer

case, we derive in Sec. A of the Supplemental Material [21]:

$$S_{\rm fp} = \lim_{N\to\infty} \frac{1}{2N} \ln \mathbb{E}\, \mathcal{N}_{\rm fp} = \frac{\ln 2}{2} + \frac{1}{2} \ln[H(-J_D - \lambda)^2 + H(-J_D + \lambda)^2], \quad (3)$$

where $H(x) = \frac{1}{2}\,\mathrm{erfc}(\frac{x}{\sqrt{2}})$. The entropy $S_{\rm fp}$ is an increasing function of J_D and λ; it vanishes when $J_D = \lambda = 0$ and goes to $\ln 2$ in the limit $J_D, \lambda \to \infty$, where all configurations become fixed points. The core model is recovered by setting $\lambda = 0$. A replica computation of the same quantity confirms that typical fixed points are isolated. According to the overlap gap property (OGP) [22], this implies that stable algorithms (which include the network dynamics as a specific case) cannot reach such fixed-point configurations in subexponential time.

While these analytical calculations correctly give the behavior of dominant configurations, they neglect the role of subdominant configurations (with an entropy smaller than $S_{\rm fp}$) that are not isolated and can be attractive for some dynamical processes and algorithms [17]. In fact, extensive simulations with several algorithms [23] show that there exist algorithm-dependent threshold values of J_D beyond which fixed points can be reached very rapidly. For the core model, belief propagation with reinforcement [18] appears to be able to find fixed points with J_D as low as 0.35 for large N in a small (sublinear) number of iterations; likelihood optimization à la Ref. [24] requires $J_D \gtrapprox 0.5$; and simple iteration of the update dynamics, on which we will focus in this paper, converges for $J_D \gtrapprox 0.8$ (see Sec. A of the Supplemental Material [21] for the scaling bound $J_D \lessapprox \sqrt{2 \ln N}$). The dynamics converge to the RM, an extended connected cluster with a high density of solutions and a branching structure [19,25]. It can be studied by a more refined analytical approach where one biases the Gibbs measure toward fixed points surrounded by a large number of neighboring solutions [17]; this "local entropy approach" is explained in the Appendix and detailed in Sec. B of the Supplemental Material [21].

The existence of the RM is a prerequisite for the convergence of local dynamical algorithms and underlies the effectiveness of our learning scheme. Its geometry combines flexibility and stability: The RM forms a dense, connected cluster of fixed points that are robust to substantial perturbations. As discussed in the Supplemental Material [21], perturbing a fixed point by flipping a large fraction of its units typically drives the dynamics toward another fixed point in its immediate neighborhood, at a Hamming distance of only a few percent. Learning further enhances this robustness by stabilizing the reached configuration, which remains stable even after the transient output feedback is removed. This persistence of task-relevant configurations provides a natural mechanism for generalization.

Moreover, the persistence of the state after the removal of the output signal, made possible by the geometrical properties of the RM, offers a simple and biologically viable protocol to implement a supervised learning scheme via local Hebbian-like weight updates, as described in the next section. This is very different from standard learning schemes that implausibly require neurons to compare their actual state with a hypothetical state reached in the absence of the supervisory signal. Alternative proposals to circumvent this issue are generally more complicated, e.g., relying on contrastive temporal comparisons [26] or more elaborate learning rules [27].

V. LEARNING

For simplicity, we shall describe our supervised learning protocol on the core module; the generalization to multilayer architectures is straightforward and can be found in the Sec. C of the Supplemental Material [21].

Assume that we are provided with a set of P patterns with their associated labels $\{x^\mu, y^\mu\}_{\mu=1}^P$ with $x^\mu \in \mathbb{R}^D$ and $y^\mu \in \{-1, 1\}^C$. We consider two projection matrices $W^{\rm in} \in \mathbb{R}^{N\times D}$, $W^{\rm back} \in \mathbb{R}^{N\times C}$ that mediate the influence of the input and label onto the network state, as well as a readout matrix $W^{\rm out} \in \mathbb{R}^{C\times N}$ that outputs a prediction given the state of the network (see Fig. 1). These matrices can be initialized and updated similarly to J.

During training, pairs x^μ, y^μ are presented one at a time. The network evolves in parallel by aligning neurons to their local field:

$$s_i \leftarrow \mathrm{sgn}\left(\sum_{j=1}^N J_{ij}\, s_j + \lambda_x \sum_{k=1}^D W^{\rm in}_{ik} x^\mu_k + \lambda_y \sum_{c=1}^C W^{\rm back}_{ic} y^\mu_c \right), \quad (4)$$

where λ_y controls whether the output influences the dynamics.

We initialize the network in a neutral state $s = 0$. Initially, only the input affects the dynamics for a few *warm-up* steps [set $\lambda_y = 0$ in Eq. (4); with L layers we use L warm-up steps]. Then, the supervision kicks in ($\lambda_y > 0$) and the network reaches a fixed point s'. From here, the transient supervision is removed ($\lambda_y = 0$) and the network reaches a second fixed point s^*. To stabilize s^*, we apply a simple local plasticity rule that increases the stability margin $s^*_i \times h_i$ of each neuron i by $\eta > 0$ if it falls below a positive threshold κ:

$$h_i = \sum_{j=1}^N J_{ij}\, s^*_j + \lambda_x \sum_{k=1}^D W^{\rm in}_k x^\mu_k, \quad (5)$$

$$J_{ij} \leftarrow J_{ij} + \eta\, s^*_i\, s^*_j\, \mathbb{1}(s^*_i \times h_i \leqslant \kappa) \mathbb{1}(J_{ij} \neq 0). \quad (6)$$

The projections $W^{\rm in}$, $W^{\rm out}$, and $W^{\rm back}$ can be learned using exactly the same principle: $W^{\rm in}$ and $W^{\rm back}$ learn to increase the margin $s^*_i \times h_i$ of each hidden neuron, while $W^{\rm out}$ learns to increase the margin of the prediction $y^\mu_c \times \hat{y}_c$, where $\hat{y} = W^{\rm out} s^*$.

Note that the training of the readout matrix $W^{\rm out}$ does not affect the learning dynamics of the rest of the network and could thus be carried out with any other regression algorithm learning to map s^* into y^μ.

During inference, the network evolves under the sole influence of the input, and upon convergence to a state $\hat{s}$ we compute the prediction $\hat{y} = W^{\rm out}\hat{s}$.

In practice, already after a few learning steps, convergence of the dynamics typically requires no more than 5–10 update steps. Furthermore, using the synchronous dynamics (4) allows an efficient vectorized implementation of our method on modern hardware, especially considering batches of data in parallel (see Sec. C of the Supplemental Material [21]).

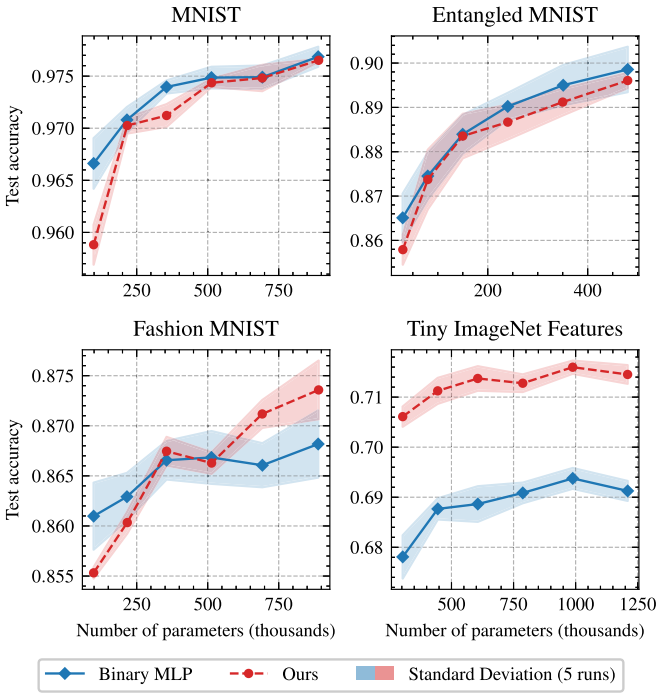

FIG. 2. Test accuracy on benchmark datasets as a function of the number of trainable parameters, controlled by the layers width. Our model (red) is compared to a binarized three-layer perceptron trained with backpropagation (blue). For each value of trainable parameters, the average test accuracy over five distinct runs is plotted, together with its standard deviation.

VI. PERFORMANCE BENCHMARKS

We benchmark the performance of our algorithm (see Sec. D of the Supplemental Material [21]) against a two-hidden-layer multilayer perceptron (MLP) trained with backpropagation, on a set of image classification benchmarks: MNIST [28], Fashion-MNIST [29], Entangled MNIST (a more challenging MNIST variant, in which MNIST features are randomly projected onto a reduced 100-dimensional space and binarized; see Sec. D of the Supplemental Material [21]), and the Tiny ImageNet subset of ImageNet [30] preprocessed by a pretrained convolutional backbone. The code to reproduce our experiments is released in Ref. [31], and raw experimental data are available from the authors upon reasonable request.

We consider the core module, fixing the sparsity coefficients to $\rho_J = 0.01$, $\rho_{W_{\text{in}}} = 0.1$, and $\rho_{W_{\text{out}}} = 1.0$ and varying the number of neurons N linearly. To ensure a fair comparison, we also vary the width of the MLP to exactly match the total number of learnable parameters. Furthermore, since our model operates with binary neurons, also the MLP employs tanh nonlinearities followed by a sign function, to binarize activations; during the backward pass, gradients are computed ignoring the sign, via straight-through estimation [32].

Figure 2 shows that our approach achieves performance comparable to that of backpropagation-based training in terms of test accuracy.

We also investigate the impact of depth in the multilayer chain model, focusing on the challenging task of heteroassociative learning (see Sec. D of the Supplemental Material [21] for task description and figures). We observe a linear dependence of the heteroassociation capacity on network width, with a coefficient that increases monotonically with depth. Furthermore, the overlap of the internal representations with the target output increases gradually with depth. We expect depth to play an even more significant role when combined with suitable inductive biases in the connectivity and leave a systematic exploration of this to future work.

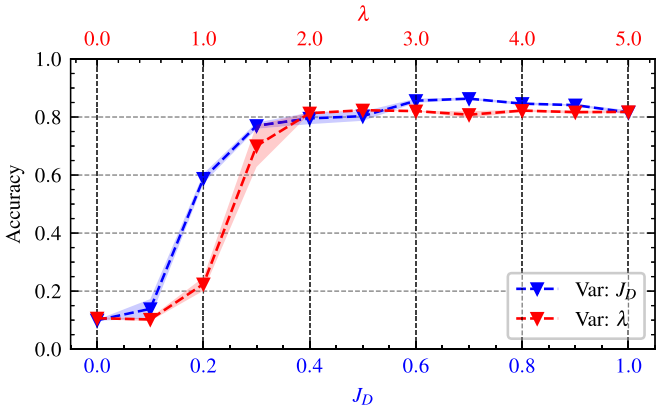

FIG. 3. Test accuracy of a two-layer chain model on Entangled MNIST as a function of λ, J_D. For each curve, one parameter is varied while the other is kept equal to zero.

Finally, we assess more closely the ability of the proposed algorithm to perform feature learning. In Sec. D of the Supplemental Material [21], we compare the performance of the core module with two simple baselines that train a linear readout on frozen features (random projection and reservoir computing [33]), matching the dimensionality of the representations. Our algorithm learns features that allow to consistently outperform these baselines in terms of training and test accuracy on Entangled MNIST, with especially pronounced gains when using low-dimensional features.

VII. ON THE ROLE OF THE RM

The importance of the RM accessibility and its geometry for learning is confirmed by several sets of experiments. In the first experiment (Sec. A of the Supplemental Material) [21], we measured how the performance of our algorithm depends on the couplings J_D and λ, using a two-layer chain model where, additionally, both layers interact with the input and the output independently. This can be thought of as two coupled and independently initialized core modules whose predictions are ensembled. We vary separately each parameter, fixing the other to zero, and we measure classification performance on the Entangled MNIST test set after a fixed number of epochs. Figure 3 shows that the network is unable to learn the task when J_D and λ are too small, but its performance suddenly and drastically improves as J_D or λ increase. Crucially, because both copies of the core module independently interact with input and output, the information propagates from the input through both layers and to the output even with $J_D = \lambda = 0$. This finding thus corroborates that the sudden performance boost is rooted in the dynamical properties of the model, and that it is directly linked with the onset of the RM.

We also study the impact of the symmetry in the coupling matrix J on performance, by considering variants of the core module that initialize J symmetrically and, optionally, use variants of the plasticity rule that preserve the symmetry (see Sec. F of the Supplemental Material [21]). With $J_D = 0$, the dynamics of the symmetric model descends a Lyapunov function and is convergent. However, it turns out that introducing and even preserving symmetry in J has no significant effect on model performance, compared to the asymmetric case. Furthermore, the symmetric variants also exhibit a drastic performance boost as J_D increases, despite the dynamics already being convergent with $J_D = 0$. This shows that the critical aspect of the RM for learning is not only its accessibility but also its geometric structure, the details of which are explored in Sec. G of the Supplemental Material [21].

VIII. CONCLUSION

We have proposed a scheme for asymmetric recurrent neural networks based on the dynamical stabilization of accessible representation manifolds. This distributed gradient-free mechanism, backed by statistical physics theory, is local both in space and in time, and it provides a viable principle for learning in recurrent neural networks, which can even compete with the state-of-the-art gradient-based algorithms.

ACKNOWLEDGMENTS

We thank Nicolas Brunel for very stimulating discussions. This work was supported by the PNRR-PE-AI FAIR project funded by the NextGeneration EU program.

The authors of this paper were ordered alphabetically.

DATA AVAILABILITY

The data that support the findings of this article are openly available [31].

APPENDIX

We explain here the "local entropy approach," an analytic method showing the existence of the RM in the system before learning takes place. It uses an effective energy density $\mathcal{E}(\tilde{s}, d)$ relative to a reference configuration of the neurons $\tilde{s}$ and the associated partition function defined as

$$\mathcal{E}(\tilde{s}, d) = -\frac{1}{N} \ln \mathcal{N}(\tilde{s}, d), \qquad Z(d, y) = \sum_{\tilde{s}} e^{-yN\mathcal{E}(\tilde{s},d)}, \tag{A1}$$

where $\mathcal{N}(\tilde{s}, d)$ is the number of fixed points s at normalized Hamming distance d from $\tilde{s}$ and the parameter y plays the role of an inverse temperature, controlling the reweighting of the local entropy contribution. The corresponding free energy density

$$\Phi(d, y) = -\lim_{N\to\infty} \frac{1}{Ny} \ln Z(d, y) \tag{A2}$$

allows to reveal subdominant regions with highest density of fixed points in the limit of large y and small d. Relevant quantities are the average local entropy, i.e., the internal entropy density of the RM, $S_I(d, y) \equiv \langle \mathcal{E} \rangle = -\frac{\partial}{\partial y}[y\Phi(d, y)]$, and the log of number of such regions, called the "external entropy density," $S_E(d, y) = -y\,(\Phi(d, y) + S_I(d, y))$. At fixed d, S_I is an increasing function of y and S_E is a decreasing function. The case $y = 1$ reduces to the computation of the entropy surrounding typical fixed points, which in our case would be the isolated ones. Considering values of $y > 1$ implies that we are exploring subdominant regions or out-of-equilibrium states.

As explained in Ref. [17], the signature of the existence of an accessible RM is that, at the largest value of y for which the external entropy $S_E(d, y) \geqslant 0$, call this $y^*(d)$, the internal entropy $S_I(d, y^*(d))$ is a monotonically growing function of d.

We evaluate Eq. (A2) in the large N limit, using the replica method under the replica symmetric (RS) ansatz. As discussed in Ref. [34], this computation can be formally interpreted as the one-step replica symmetry breaking calculation of the original problem of computing the entropy of fixed points, where now the Parisi parameter m must be identified with y, and the internal overlap q_1 is used as a control parameter to fix the region's diameter via $d = (1 - q_1)/2$.

The asymptotic expression for the free energy and related entropies can be derived for an arbitrary number of layers (see Sec. B of the Supplemental Material [21]). However, solving the saddle-point equations poses difficult numerical challenges, and we limit our analytical study to the case of two layers, which is enough to display the role of both J_D and λ. The expression for the free energy density reads

$$\begin{aligned} -y\Phi(d, y) &= -\hat{q}_1 + (1 - y)\,\hat{q}_1 q_1 + \frac{1}{y} \ln \left\{ \int Dz\, Dz' Dt\, Dt' \right. \\ &\times \left[\sum_{s,s' \in \{\pm 1\}} e^{\sqrt{\hat{q}_1}(st + s't')} H\left(\frac{-J_D + \lambda\, ss' - s\sqrt{q_1}z}{\sqrt{1 - q_1}} \right) \right. \\ &\left. \left. \times H\left(\frac{-J_D + \lambda\, ss' - s'\sqrt{q_1}z'}{\sqrt{1 - q_1}} \right) \right]^y \right\}, \end{aligned} \tag{A3}$$

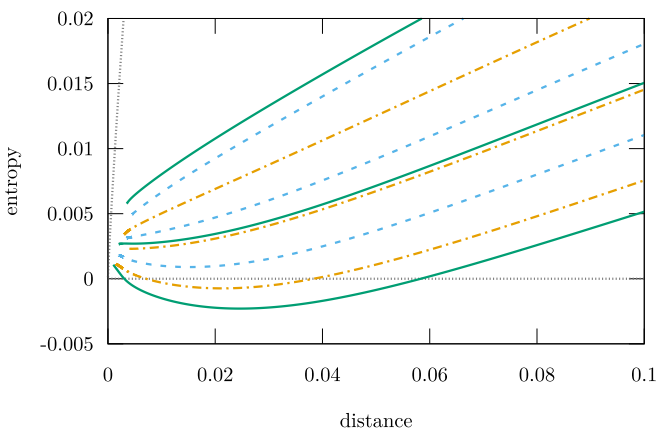

FIG. 4. Internal entropy (divided by L) for the core model (solid green) and two-layer model (dashed/dot-dashed blue/orange). Core model: $J_D = 0.05, 0.07, 0.1$ (bottom to top). Two-layer model: blue ($J_D = 0.05$) with $\lambda = 0.2, 0.3, 0.4$ (bottom to top); orange ($J_D = 0$) with $\lambda = 0.4, 0.45, 0.5$ (bottom to top). All curves vanish at distance 0 (not shown due to numerical limits). In all cases, increasing J_D or λ drives the OGP-RM transition.

where Dz, Dz', Dt, Dt' are Gaussian measures with mean 0 and variance 1, and $\hat{q}_1$ is determined by the saddle-point equation $\partial\Phi/\partial\hat{q}_1 = 0$ (see Sec. B of the Supplemental Material [21]). The results for the core and multilayer chain models are as follows.

Consider the single core module. The free energy density is recovered by setting $\lambda = 0$ and dividing by 2 (to account for the number of neurons). Solving the saddle-point equations reveals that, for all $J_D > 0$ and d, there exists a value of y beyond which the external entropy $S_E(d, y)$ becomes negative. This is unphysical and indicates a breakdown of the RS assumption, signaling a phase transition. To identify the critical values of J_D, we first compute, for each value of d, the value y^* at which the external entropy vanishes, and next seek the value of J_D below which the internal entropy becomes a nonmonotonic function of the distance. We find that the phase space separates in two regions. For all $J_D < J_{Dc} \simeq 0.07$, the internal entropy is nonmonotonic. The value of $d = (1 - q_1)/2$ that maximizes the entropy coincides with the self-overlap of subdominant states, given that the associated value of y^* exceeds one. The actual values of d are extremely small, suggesting that finding fixed points belonging to such states would still be hard for local algorithms. For all $J_D > J_{Dc}$, the internal entropy increases monotonically with d, eventually saturating at a plateau equal to the entropy of typical solutions. This behavior indicates the existence of an extensive cluster in configuration space where fixed points coalesce giving rise to the RM geometry.

The same analysis can be extended to the multilayer model. Focusing on the case of $L = 2$, we find that the role of λ closely parallels that of J_D: A positive sparse coupling λ between different subsystems induces the emergence of an RM. Specifically, for small values of J_D, including zero, a critical value of $\lambda \lessapprox 0.45$ exists at which the extended high-density cluster of fixed points emerges. Figure 4 shows the behavior of the local entropy for the two cases.

[1] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[2] The new NeuroAI Panel, The new NeuroAI, Nat. Mach. Intell. **6**, 245 (2024).

[3] A. Ororbia, A. Mali, A. Kohan, B. Millidge, and T. Salvatori, A review of neuroscience-inspired machine learning, arXiv:2403.18929.

[4] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in *International Conference on Machine Learning* (PMLR, Brookline, MA, 2015), pp. 2256–2265.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, NY, 2017), Vol. 30, pp. 5998–6008.

[6] D.-H. Lee, S. Zhang, A. Fischer, and Y. Bengio, Difference target propagation, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, Cham, 2015), pp. 498–515.

[7] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, Random synaptic feedback weights support error backpropagation for deep learning, Nat. Commun. **7**, 13276 (2016).

[8] R. P. Rao and D. H. Ballard, Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects, Nat. Neurosci. **2**, 79 (1999).

[9] B. Millidge, A. Seth, and C. L. Buckley, Predictive coding: A theoretical and experimental review, arXiv:2107.12979.

[10] G. Hinton, The forward-forward algorithm: Some preliminary investigations, arXiv:2212.13345.

[11] K. Nakajima and I. Fischer, *Reservoir Computing* (Springer, Cham, 2021).

[12] J. H. Lee, T. Delbruck, and M. Pfeiffer, Training deep spiking neural networks using backpropagation, Front. Neurosci. **10**, 508 (2016).

[13] Z. Yu, W. Huang, and H. Huang, Neural Langevin machine: A local asymmetric learning rule can be creative, arXiv:2506.23546.

[14] B. Scellier and Y. Bengio, Equilibrium propagation: Bridging the gap between energy-based models and backpropagation, Front. Comput. Neurosci. **11**, 24 (2017).

[15] H. Sompolinsky, A. Crisanti, and H.-J. Sommers, Chaos in random neural networks, Phys. Rev. Lett. **61**, 259 (1988).

[16] M. Stern, H. Sompolinsky, and L. F. Abbott, Dynamics of random neural networks with bistable units, Phys. Rev. E **90**, 062710 (2014).

[17] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Subdominant dense clusters allow for simple learning and high computational performance in neural networks with discrete synapses, Phys. Rev. Lett. **115**, 128101 (2015).

[18] C. Baldassi, C. Borgs, J. T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes, Proc. Natl. Acad. Sci. USA **113**, E7655 (2016).

[19] C. Baldassi, C. Lauditi, E. M. Malatesta, G. Perugini, and R. Zecchina, Unveiling the structure of wide flat minima in neural networks, Phys. Rev. Lett. **127**, 278301 (2021).

[20] D. Barbier, Finding the right path: Statistical mechanics of connected solutions in constraint satisfaction problems, arXiv:2505.20954.

[21] See Supplemental Material at http://link.aps.org/supplemental/10.1103/lbr6-91sn for detailed analytical derivations, extended learning algorithms, and additional experimental results, which includes Refs. [35–38].

[22] D. Gamarnik, The overlap gap property: A topological barrier to optimizing over random structures, Proc. Natl. Acad. Sci. USA **118**, e2108492118 (2021).

[23] D. Badalotti, C. Baldassi, M. Mezard, M. Scardecchia, and R. Zecchina, Dynamical learning schemes for deep learning, in preparation (unpublished).

[24] C. Baldassi, F. Gerace, H. J. Kappen, C. Lucibello, L. Saglietti, E. Tartaglione, and R. Zecchina, Role of synaptic stochasticity in training low-precision neural networks, Phys. Rev. Lett. **120**, 268103 (2018).

[25] B. L. Annesi, C. Lauditi, C. Lucibello, E. M. Malatesta, G. Perugini, F. Pittorino, and L. Saglietti, Star-shaped space of solutions of the spherical negative perceptron, Phys. Rev. Lett. **131**, 227301 (2023).

[26] L. Saglietti, F. Gerace, A. Ingrosso, C. Baldassi, and R. Zecchina, From statistical inference to a differential learning rule for stochastic neural networks, Interface Focus **8**, 20180033 (2018).

[27] A. Alemi, C. Baldassi, N. Brunel, and R. Zecchina, A three-threshold learning rule approaches the maximal capacity of recurrent neural networks, PLoS Comput. Biol. **11**, e1004439 (2015).

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (1998).

[29] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms, arXiv:1708.07747.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2009), pp. 248–255.

[31] D. Badalotti, M. Scardecchia, C. Baldassi, R. Zecchina, and M. Mezard, Willinki/darnax: v0.6.1, Zenodo (2026), https://doi.org/10.5281/zenodo.19496650.

[32] Y. Bengio, N. Léonard, and A. Courville, Estimating or propagating gradients through stochastic neurons for conditional computation, arXiv:1308.3432.

[33] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks—with an erratum note, Technical Report 148, German National Research Center for Information Technology, 2001.

[34] C. Baldassi, F. Pittorino, and R. Zecchina, Shaping the learning landscape in neural networks around wide flat minima, Proc. Natl. Acad. Sci. USA **117**, 161 (2020).

[35] K. He, X. Zhang, S. Ren, and J. Sun, Deep Residual Learning for Image Recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), pp. 770–778.

[36] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, RandAugment: Practical data augmentation with no separate search, in *Advances in Neural Information Processing Systems* (2020), Vol. 33, pp. 16754–16765.

[37] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, CutMix: Regularization strategy to train strong classifiers with localizable features, in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, 2019), pp. 6023–6032.

[38] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, mixup: Beyond Empirical Risk Minimization, in *International Conference on Learning Representations* (ICLR, 2018).