Waddington landscape for prototype learning in generalized Hopfield networks

Nacer Eddine Boukacem
Département de Biochimie et Medecine Moléculaire, Université de Montréal, Montréal, Quebec H3C 3J7, Canada
and Rutherford Physics Building, McGill University, Montréal, Quebec H3A 2T8, Canada

Allen Leary
Rutherford Physics Building, McGill University, Montréal, Quebec H3A 2T8, Canada
and Regeneron Pharmaceuticals, 777 Old Saw Mill River Road, Tarrytown, New York 10591, USA

Robin Thériault
Rutherford Physics Building, McGill University, Montréal H3A 2T8, Quebec, Canada
and Scuola Normale Superiore di Pisa, Piazza dei Cavalieri, 7 - 56126 Pisa, Italy

Felix Gottlieb
Rutherford Physics Building, McGill University, Montréal H3A 2T8, Quebec, Canada

Madhav Mani
Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, Illinois 60208, USA
and NSF-Simons Center for Quantitative Biology, Northwestern University, Evanston, Illinois 60208, USA

Paul François[*]
Département de Biochimie et Medecine Moléculaire, Université de Montréal, Montréal, Quebec H3C 3J7, Canada
and MILA Québec, Montréal, Quebec H2S 3H1, Canada

(Received 11 January 2024; accepted 21 June 2024; published 23 July 2024)

Networks in machine learning offer examples of complex high-dimensional dynamical systems inspired by and reminiscent of biological systems. Here, we study the learning dynamics of generalized Hopfield networks, which permit visualization of internal memories. These networks have been shown to proceed through a "feature-to-prototype" transition, as the strength of network nonlinearity is increased, wherein the learned, or terminal, states of internal memories transition from mixed to pure states. Focusing on the prototype learning dynamics of the internal memories, we observe stereotypical dynamics of memories wherein similar subgroups of memories sequentially split at well-defined saddles. The splitting order is interpretable and reproducible from one simulation to the other. The dynamics prior to splits are robust to variations in many features of the system. To develop a more rigorous understanding of these global dynamics, we study smaller subsystems that exhibit similar properties to the full system. Within these smaller systems, we combine analytical calculations with numerical simulations to study the dynamics of the feature-to-prototype transition, and the emergence of saddle points in the learning landscape. We exhibit regimes where saddles appear and disappear through saddle-node bifurcations, qualitatively changing the distribution of learned memories as the strength of the nonlinearity is varied—allowing us to systematically investigate the mechanisms that underlie the emergence of the learning dynamics. Several features of the learning dynamics are reminiscent of the Waddington's caricature of cellular differentiation, and we attempt to make this analogy more precise. Memories can thus differentiate in a predictive and controlled way, revealing bridges between experimental biology, dynamical systems theory, and machine learning.

DOI: 10.1103/PhysRevResearch.6.033098

I. INTRODUCTION

The field of machine learning offers fascinating examples of networks with self-organizing dynamics. During learning, network parameters change and qualitatively novel regimes appear [1,2]. The dynamics through qualitatively distinct learning regimes open up new possibilities for their optimization and, thus, training. Task optimization in very high dimensions is now seen as "easy" for two reasons: first, as

[*]Contact author: paul.francois@umontreal.ca

the high dimensionality of the networks generally ensures that at least one eigenvalue is negative at critical points—there is always a direction in which the loss can be further reduced [3] and second, local minima in very high dimensions turn out to be very good at generalizing [3,4]. This suggests that only saddle points (or saddles) are met during optimization driven by gradient descent [5], so that the learning dynamics consist of transitions between saddles (learning can be further sped up using various standard methods). However, the quantitative and qualitative natures of the saddles themselves might be irreproducible, depending on details of the system. Such a view of learning, with potentially a huge number of (random) critical points [3], contrasts the statistical regularities observed during the learning dynamics of neural networks. To this end, in the limit of high dimensionality, exact results on the biases and errors of learning dynamics have been derived using tools inspired by random matrix theory and statistical mechanics [6,7]. In addition, information theory based metrics have further highlighted the structured and reproducible dynamics during learning [8,9].

Together, these studies suggest the emergence of potentially universal, and thus, simpler, regimes of learning, echoing what is observed in other self-organizing dynamical systems, in particular in biological contexts [10,11]. Case in point: The advances in high throughput quantitative data in biology combined with mathematical modeling have established that the dynamics of high-dimensional biological networks can often be captured by low-dimensional representations, at least locally, see e.g., [12] for cell mechanics or [13] for brain decisions. An important example is cellular differentiation [14], for which Conrad Waddington introduced the qualitative notion of an "epigenetic landscape" to describe low-dimension dynamics. In Waddington's picture, the evolution of cellular states is represented by a ball rolling down "valleys" accounting for cellular states in an abstract space [15]. Valleys can split, leading to differentiation event, and are "canalized", i.e., are robust to perturbations of the system [16]. Waddington's landscape caricature has motivated multiple experimental studies, verifying some of its most salient features—for instance, we know now that cellular differentiation occurs through progenitor states, consistent with Waddington's valleys [17]. Multiple attempts have been made to mathematically characterize such landscapes. In pioneering study, "classical" Hopfield networks themselves have been used to reverse-engineer an epigenetic landscape [18], and further revealed a 1D reaction coordinate during cellular reprogramming [19]. The nature of binary decisions within (biological) landscapes can be rigorously classified between binary choice or binary flip [14], leading to predictions and applications in specific systems [20]. While those explicit approaches have met success, it remains unclear how and why low dimensional dynamics emerge from the complex interacting components that comprise a typical differentiation network [10,21]?

Here, we provide a perspective, by characterizing hierarchical, low-dimensional dynamics reproducibly emerging in a class of machine learning algorithms. We focus on generalized Hopfield networks (GHN), an architecture that has been suggested to be capable of capturing multiple machine learning frameworks such as large language [2] or diffusion models [22]. It has been noted that GHNs can work in two distinct regimes [23,24], characterized by two hyperparameters (n, T_r), which loosely correspond to the degree of nonlinearity and an effective temperature, respectively. In our present study, going beyond the trained state of the system, we establish that for higher n the learning dynamics become lower dimensional, with convergence of learning trajectories to well-defined, reproducible, and interpretable saddles (in a dynamical system sense [25]), followed by binary splitting events leading to increased specialization/differentiation of internal memories. We study simpler versions of such networks with similar properties, for which we present analytical results that provide specificity and rigor to our claims. Taken as a whole, despite the manifest literal differences in the parts that make up the process, the dynamics we observe in GHNs bear a close resemblance to Waddington's caricature of cellular differentiation, suggesting that they might be generic features of self-organizing complex systems [26].

II. RESULTS

A. Generalized Hopfield networks for classification

Generalized Hopfield networks were introduced by Krotov and Hopfield in two seminal papers [23,24]. They represent an elaboration of the classical Hopfield model for associative memory [1], one of the first modern neural network architectures designed to perform complex tasks. In brief, generalized Hopfield networks rely on a well-designed, spin-glass type, energy function, allowing to (1) store and (2) recover patterns. A generalized energy can be first defined,

$$H(|\sigma\rangle) = -\sum_{\mu} f_n\left(\frac{\langle M_\mu|\sigma\rangle}{T}\right). \tag{1}$$

In this expression, $|\sigma\rangle$ corresponds to a system configuration. We use the standard ket notation to indicate a vector, and the bra-ket $\langle\cdot|\cdot\rangle$ notation for dot product. The stored internal patterns correspond to vectors $|M_\mu\rangle$ that we call "memories" where μ is the index of a memory considered. The memories have the same dimension as the input, so for instance, if $|\sigma\rangle$ corresponds to a picture, the memories $|M_\mu\rangle$ can themselves be interpreted as pictures, and corresponding pixels take values in $[-1, 1]$. Dot products between pictures and memories are divided by a "temperature", T. T should scale with the dimensionality of the space considered, for this reason, we introduce a rescaled temperature $T_r = \frac{T}{D}$, where D is the dimension of the memory-space (e.g., for 28×28 pixel pictures, $D = 784$). T_r is a hyperparameter of the network, which is fixed for one given learning simulation, and is typically of order 1. Finally, a nonlinear function, whose degree is determined by the magnitude of n, is applied to each dot product

$$f_n(x) = (ReLU(x))^n = \begin{cases} x^n, & x \geqslant 0 \\ 0, & x < 0 \end{cases}. \tag{2}$$

In the original Hopfield picture, patterns correspond to energy minima, and can be recovered from a more or less noisy initial condition through gradient descent, iteratively changing $|\sigma\rangle$ to minimize the energy H. Contrasting the

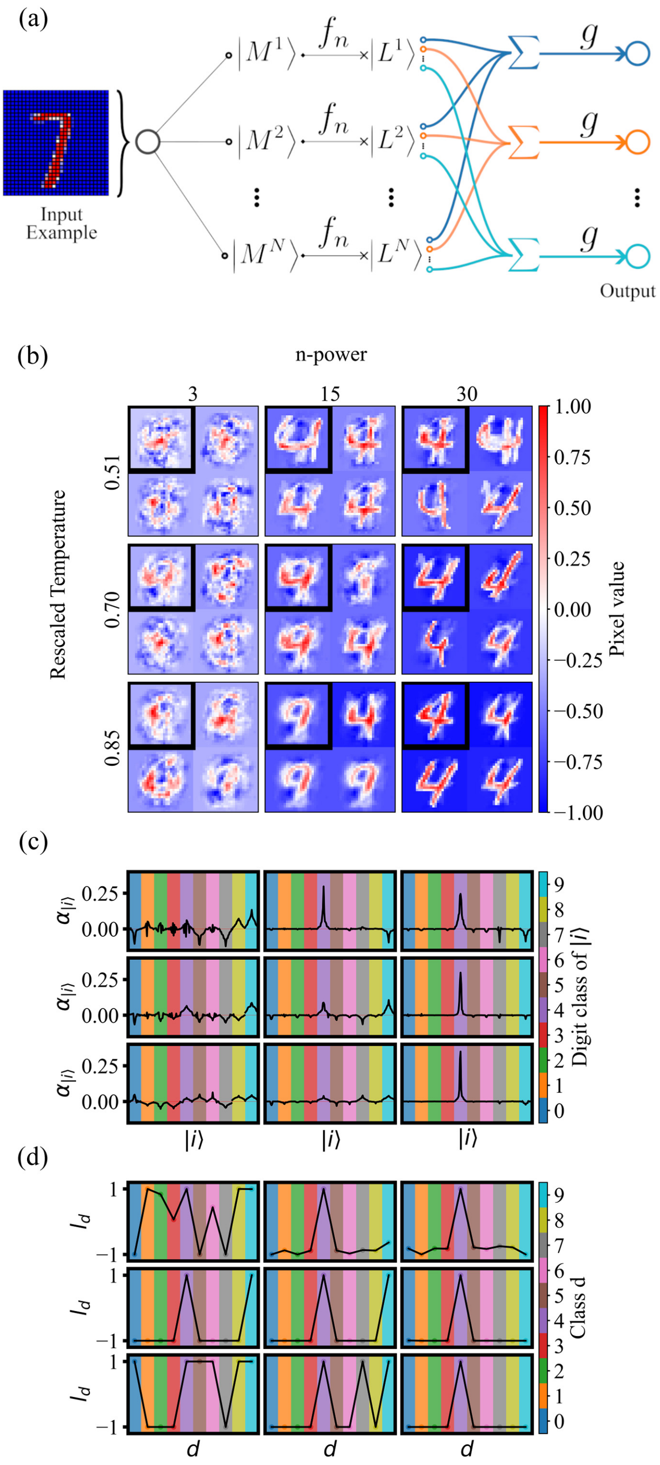

FIG. 1. Feature-prototype transition for generalized Hopfield networks. (a) Illustrates the general architecture used. For a given (n, T_r), (b) shows samples of the memory final states. The memories chosen are all associated with a final label contributing highly ($\geqslant 0.99$) to the class 4. The top-left memory for each (n, T_r) is then linearly decomposed using the pseudo-inverse of the training set and in (c), the contribution of each training example in that memory is plotted. These linear coefficients are symmetrically reordered such that for each class, the maximum is at the center. In (d), the label associated to this same memory is shown. For all simulations shown in this figure, the network size was 100 memories, a training rate of 0.005 and random gaussian initial conditions. The training set was identical for all runs and consisted of 200 MNIST digits (20 of each class). The simulations vary only in the temperature $T_r = (0.51, 0.7, 0.85)$ and $n = (3, 15, 30)$.

original Hopfield network, a hyperparameter n is introduced into the GHN, which, intuitively, is expected to "steepen" the energy wells around "true" minima (corresponding to memories in Hopfield's initial picture), thus possibly leading to less spurious minima and more efficient learning/encoding. Indeed, it is now well established that such steepening allows for more "packing" of information in memory space, with an explosion of memory capacity for exponential versions of generalized Hopfield networks [27], which rationalizes the current renewed interest in them.

In [23,24] Krotov and Hopfield proposed to modify the generalized Hopfield framework and instead train a classifier using a similar energy function as an intermediate layer, see Fig. 1(a) for an illustration of the architecture. They consider a model with a hidden layer of (internal) memories $|M^\mu\rangle$ and define the output, $|o(\sigma)\rangle$, of the network in response to a sample,

$$|o(\sigma)\rangle = g\left(\sum_\mu |L_\mu\rangle\, f_n\left(\frac{\langle M_\mu|\sigma\rangle}{T}\right)\right). \tag{3}$$

The output $|o(\sigma)\rangle$ is a vector used for classification: each element of this vector can be understood as a score (between -1 and 1) quantifying if $|\sigma\rangle$ belongs (or not) to a given class [there are as many possible classes as the dimension of $|o(\sigma)\rangle$]. In the expression above, each internal memory $|M_\mu\rangle$ is associated to a label vector $|L_\mu\rangle$ of same dimensionality as $|o(\sigma)\rangle$, quantifying the proximity of memory $|M_\mu\rangle$ to each predefined class—thereby performing a classification task. g is a nonlinear function, applied elementwise, and typically one takes $g = \tanh$. The parameters of the networks thus are its internal memories $|M_\mu\rangle$ and their vectorized labels or classes $|L_\mu\rangle$.

To train the classifier, a cost function is encoding categorical accuracy, i.e., imposing that the vector $|o(\sigma)\rangle$ is as close as possible to its "true" value $|t_{|\sigma\rangle}\rangle$ encoding the class of $|\sigma\rangle$,

$$C = \sum_{|\sigma\rangle\in\mathcal{T}} \sum_d (o_d(|\sigma\rangle) - t_d(|\sigma\rangle))^{2m}. \tag{4}$$

In the above expression o_d and t_d correspond to the dth coordinate of the respective vectors $|o(\sigma)\rangle$ and $|t_{|\sigma\rangle}\rangle$, and $\mathcal{T}$ is the training ensemble {we can train/evaluate the model either with minibatches/subsets or with the entire training set without changing the qualitative results, see details in Sec. 1.3.3 and Figs. S.5–S.8 within the Supplemental Material (SM) [28]}. We typically use $m = n$ like in the original study.

To be more concrete, in the specific classification task of the MNIST dataset [29] of handwritten digits, $|\sigma\rangle$ is a picture of a digit, and $|o(\sigma)\rangle$ a ten-dimensional vector, corresponding to the ten digit classes $(0, 1, 2, ..., 9)$. So, if say $|\sigma\rangle$ represents a 1, ideally the corresponding output vector should be $|o(\sigma)\rangle = |t_{|\sigma\rangle}\rangle = (-1, +1, -1, \dots, -1)$. During learning, parameters are changed through gradient descent of the cost function. A rescaling condition is also imposed to make sure that all of the pixels in the memories are bound between $[-1, 1]$. Equations (S.16) and (S.17) (see the SM [28]) provide the actual expressions used to compute the derivatives of the cost function with respect to parameters, including contributions from the cost functions, the labels $|L_\mu\rangle$ and the rescaling, which is then used for gradient descent. Importantly, each internal memory $|M_\mu\rangle$ is updated at each epoch with small contributions coming from all training samples (with individual weights depending on the cost function), meaning that the state of internal memories, $|M_\mu\rangle$, is a linear combination of all samples in the training set. Additional technical aspects of the model and training are presented in the Sec. 1.3 within the SM [28]. All code used for the simulations and for the figures is available at the GitHub repository [30].

Krotov and Hopfield tested their architecture on the MNIST dataset [29] of handwritten digits. They studied the hidden memories at the end of training, and visually observed a striking change in the nature of terminal internal memories $|M_\mu\rangle$ as the hyperparameter n is increased, as illustrated in Fig. 1. For low n, the internal memories look like overlaps of multiple digits Fig. 1(b) top left, which suggests an encoding that is distributed across memories. Conversely, for high n, the internal memories look like actual digits from the training samples, suggesting an encoding based on proximity to exemplar—or prototype—digits. Based on those observations, they proposed that digit classification transitions from a "feature-based" encoding to a "prototype-based" encoding, Figs. 1(b)–1(d). This transition in the encoding was not quantified, although they also showed that the number of internal memories contributing to the classification of a specific digit considerably decreases with increasing n [23]. This confirms that in the prototype regime, input samples are effectively compared to only a few internal memories to achieve correct classification. For this reason, prototype-based classification is of particular interest since it is more understandable and interpretable. Furthermore, for high n, Krotov and Hopfield demonstrated the scheme's robustness to adversarial perturbations specifically designed to fool the classifier [24].

B. Characterizing feature-to-prototype transitions using Moore-Penrose pseudo-inverse

Figures 1(a)–1(d) illustrate the feature-to-prototype transition in GHNs, using a small training ensemble of 200 digits, which allows us to visualize what happens. As said above, because of the gradient descent training, each memory $|M\rangle$ can be decomposed into a linear sum of samples used in the training set. Formally, we write for each memory (indexed by μ)

$$|M^\mu\rangle = \sum_{|i\rangle\in\mathcal{T}} \alpha^\mu_{|i\rangle}\, |i\rangle \tag{5}$$

where $\mathcal{T}$ defines the entire training set, $|i\rangle$ is a generic label for the ith vector in the training set, and $\alpha^\mu_{|i\rangle}$ a corresponding weight for $|M^\mu\rangle$. Given a memory $|M^\mu\rangle$ and a training set, we can compute such a decomposition using the Moore-Penrose pseudo-inverse, which can be derived from the singular value decomposition of the matrix where line i corresponds to vector $|i\rangle$ (see more details in Sec. 2 within the SM [28]). The $\alpha^\mu_{|i\rangle}$ are not unique if the vectors in the training set are not linearly independent; however, since MNIST has dimension $784 = 28 \times 28$ (pixels), we do not expect any ambiguity with the computation of the $\alpha_{|i\rangle}$ if the training set has much fewer than 784 elements (e.g., 200). We refer to Fig. S.1 within the SM [28] for illustrations of the reconstruction.

In Fig. 1(b) we show a sample of memories $|M^\mu\rangle$ at the end of the training such that $l^\mu_{d=4} \geqslant 0.99$, i.e., memories contributing to classifying inputs as 4, for various n and rescaled temperature T_r. We also show the distribution of $\alpha_{|i\rangle}$ in Fig. 1(c) for the nine memories highlighted in black in Fig. 1(b) (top left corner for each n, T_r pair). The $|i\rangle$ axis represents the 200 samples ordered per digit, and we plot the corresponding $\alpha_{|i\rangle}$. To better see the distributions of $\alpha_{|i\rangle}$, for each digit we reordered the $|i\rangle$ so that the distribution looks symmetrical, which allows us to get an intuitive sense of what happens for each memory and each digit. Figure 1(d) illustrates the corresponding l^μ_d as a function of digit category d for the same memories.

Looking at Figs. 1(b)–1(d), we thus see that for low n, we recover "feature"-like memories. They consist of positive and negative linear combinations of many different digits with relatively small weights [see e.g., top-left corner of Fig. 1(c)], explaining their disordered appearance. Interestingly, these features are not random, e.g., as n is increased, similar digits (e.g., typically 4, 7, 9, see below) usually contribute significantly positively or negatively. Labels for positive digits typically take the maximum value of 1 while other labels are -1 Fig. 1(d). As both n and T_r are increased, fewer input samples contribute to the memories, giving more peaked distributions of $\alpha_{|i\rangle}$, Fig. 1(c), until one gets a very peaked distribution with very few inputs, associated with the same digit. This gives rise to well-defined prototypes, e.g., bottom-right corner in Fig. 1(b), and correspondingly only one label is positive for larger n, T_r, Fig. 1(d).

We evaluated a few more metrics related to the $\alpha_{|i\rangle}$ to quantitatively characterize the transition. Figure 2(a), left, shows the maximum value of $\alpha_{|i\rangle}$ at the end of the training, averaged over all memories in the system. We see at least two regimes. For smaller n, the maximum α is small, indicating that no input sample dominates, and thus a very distributed encoding. As n increases, there is a threshold in n, *with possibly a discontinuous derivative*, from which the maximum α increases to significantly high values, thus defining prototypes. In Fig. 2(a), right panel, we further show the average number of training samples necessary to reconstruct memories up to a small tolerance (see details in Sec. 7.1 within the SM [28]). Consistent with the behaviors on αs, we see an approximately linear decrease for rescaled temperatures $0.57, 0.83$, up to a low plateau, where few samples define the memory. Surprisingly, when the rescaled temperature gets closer to 1, we observe an intermediate plateau for intermediate n, where on average about 50 to 100 samples are necessary to reconstruct a given memory. To provide an alternate visualization, and inspired by computational biology (e.g., the analysis of single-cell RNA seq data [31,32]), we show memory samples as well as memory locations using a UMAP embedding [33,34] of the MNIST dataset, Fig. 2(b) (see details on all UMAP embeddings used in Sec. 7.2 within the SM [28]). While for high n the memories are well spread in each cluster of the UMAP embedding, for smaller n and higher T_r the memories are

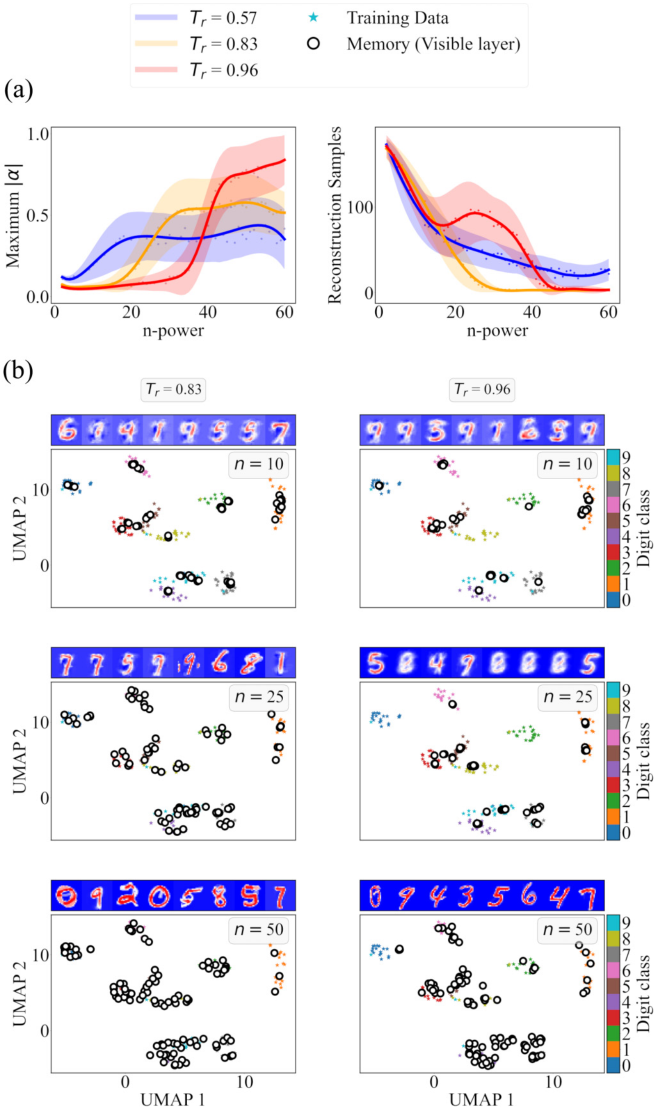

FIG. 2. (a) Average of maximum α and average number of prototypes needed to reconstruct each memory, as a function of n at the end of training, for different temperatures. (b) Representation of the learned memories for various n, T_r using a common UMAP 2D embedding, computed with the entire MNIST training database. The training dataset is represented with points, colored according to their class, see the colorbar on the right. The memories at the end of training are represented by white disks, all learnt memories (100) are shown. A small sample of memories obtained is shown above each plot. For higher n, the memories are more spread out and completely cover each cluster, indicating good coverage of the training set used. Other simulation parameters are the same as Fig. 1.

typically more concentrated in the UMAP embedding, close to the center of each cluster, indicating that there typically is less variety in the memories.

C. Hierarchical learning dynamics in the prototype regime

The striking simplicity of the learning outcome we report in the prototype regime motivates a more in-depth study of the internal memories to investigate when and how memories converge. We first follow and interpret the learning dynamics by visualizing the memories as a function of training epochs. We visualize learning using the UMAP embedding of the MNIST dataset defined in the previous section [see Fig. 3(a) for $n = 3$, feature mode; and Fig. 3(e) $n = 30$, prototype mode, $T_r = 0.85$], (see also Movies 1 and 2, as well as Figs. S.3 and S.4 within the SM [28] for other values of n). We also quantified how memories move between UMAP clusters using transition matrices, Figs. 3(b) and 3(f) (see Sec. 7.3 within the SM [28]). We show samples of memories at different epochs, ordered from 0 to 9 based on their dominant labels l_d^μ at the end of the training Figs. 3(c) and 3(g). We finally show the number of digits correctly recognized as a function of the epochs in Figs. 3(d) and 3(h).

The learning dynamics in simulations for $n = 3$ appear rather uniform, in the sense that the memories distribute themselves and “diffuse” simultaneously across most digits as the number of epochs increases, Figs. 3(a)–3(d). (Notice, however, that the UMAP embedding is difficult to interpret for small n since memories do not necessarily correspond to well-defined digits.) Memories change between almost all clusters, as can be seen in both Fig. 3(a) and the transition matrix Fig. 3(b). After an initial period where only digits of category 1 are correctly recognized, around epoch 370 subsets of almost every digit are correctly classified, and the number of properly classified digits in each category simply increases from there, Fig. 3(d).

By contrast, the training dynamics for $n = 30$ favour some specific digits at different epochs of learning, with clear sequential steps Figs. 3(e)–3(g). The transitions between clusters appear to follow a well-defined tree in the UMAP space Fig. 3(e), and the transition matrix Fig. 3(f) is sparse, indicating that there are “preferred” transitions between clusters. Figure 3(g) further illustrates the learning dynamics. All internal memories are initially similar and quickly converge towards a memory resembling a 1. Subsequently, new digits (and corresponding memories) are sequentially learned. The digits 4 are initially learned around epoch 1000, then 9, 7 and 5, 6 around epoch 1300, and other digits later. The order of appearance of memories is consistent with the order in which digits are properly recognized, Fig. 3(h).

We show multiple other examples of training in Figs. S.3–S.16 within the SM [28]. In particular, while there is some variability depending on initial conditions and hyperparameters (compare, e.g., Fig. 3 and Figs. S.13 and S.15 within the SM [28]), the sequence of digits that are learned is reproducible from one simulation to another as n is increased, and pairs of digits resembling one another are learned together. In Fig. S.17 within the SM [28], we illustrate the average order of appearance over 100 simulations, each with 200 memories, with the same parameters as Fig. 3(e). Typically, 1, 9 are learnt quickly first followed by 4, 7; then 6, 5; 8, 3 and lastly 0 and 2.

Looking in more detail at individual memories, the learning process at high n is characterized by the following stereotyped sequence of events: an ensemble of memories that are initially identical, converge to a well-defined state, then the symmetry is broken wherein the initial ensemble divides in two sub-groups. This explains the “tree-like” structures observed in the UMAP embedding Fig. 3(e) and the sparse transition matrix Fig. 3(f). More specifically, focusing on the

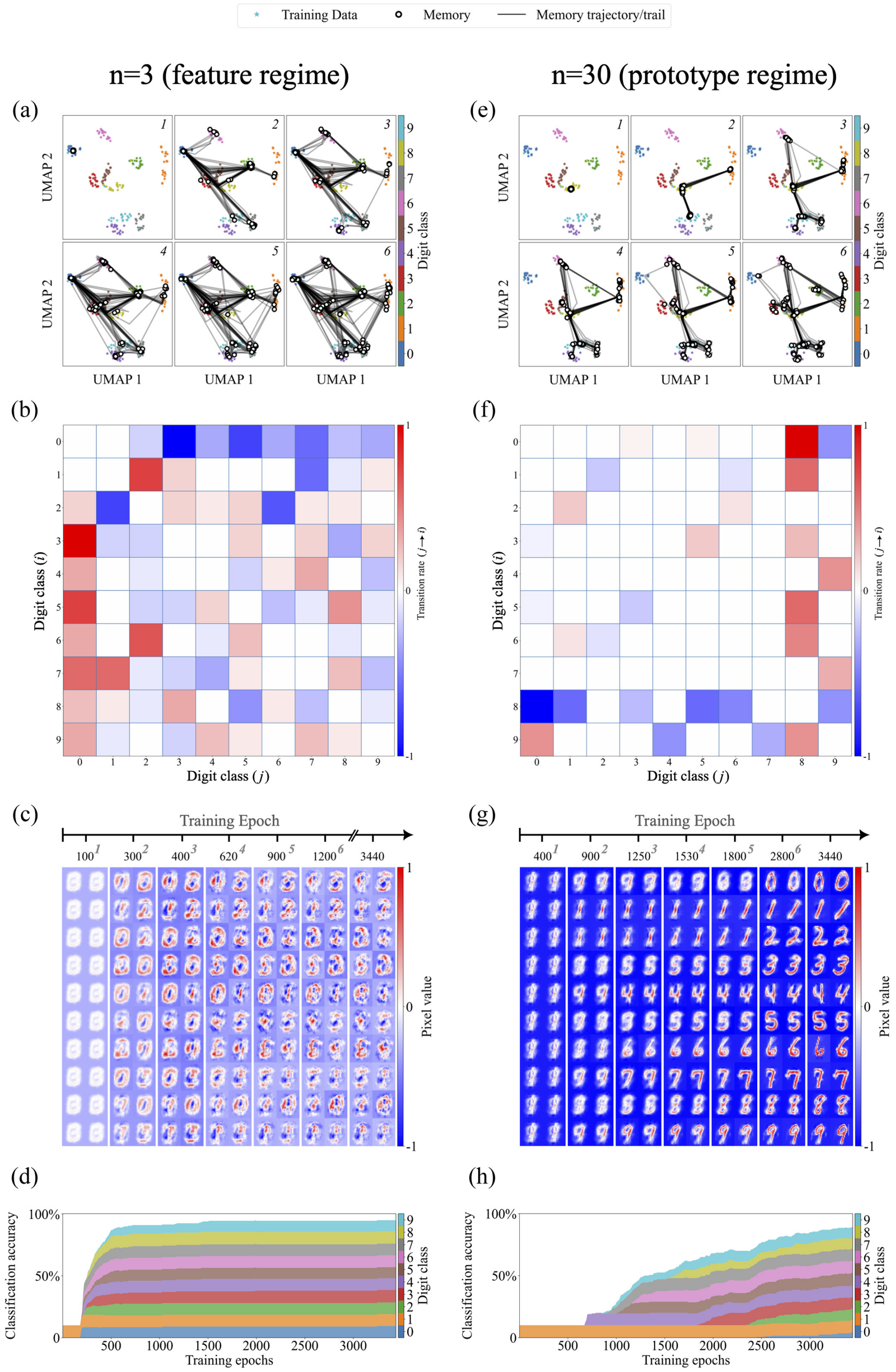

FIG. 3. Illustration of the learning dynamics, for two different values of n. Panels (a)–(d) illustrate what happens for $n = 3$ (feature mode), while (e)–(h) are for $n = 30$ (prototype mode). (a) The memories are transformed onto the UMAP latent space and plotted according to the epoch of training (indicated at top-right corner of each subplot). Gray lines indicate transitions between clusters, defining learning trajectories. When the same transition occurs many times, gray lines accumulate, so that gray/black levels visually correlate to the frequency of transitions [see panel (b)] The trajectories start at epoch 100 and end at the "current epoch" of each subplot. (b) Transition rates during learning between UMAP clusters corresponding to different digits are shown in matrix form, blue corresponds to negative fluxes and red to positive fluxes. For instance, here many memories move from 0 cluster to 2–9 clusters, so that the 0 cluster is a central hub for all transitions. Notice there are fluxes between almost all clusters (c) a subset of the memories through training are shown. For the simulation behind this figure, $T_r = 0.85$, the training set and other parameters are similar to Fig. 1. (d) Each coloured band height is proportional to the number of digits properly recognized as a function of the epochs, with color code indicated on the right. (e)–(g) Equivalent figures to (a)–(d) for $n = 30$ (prototype mode).

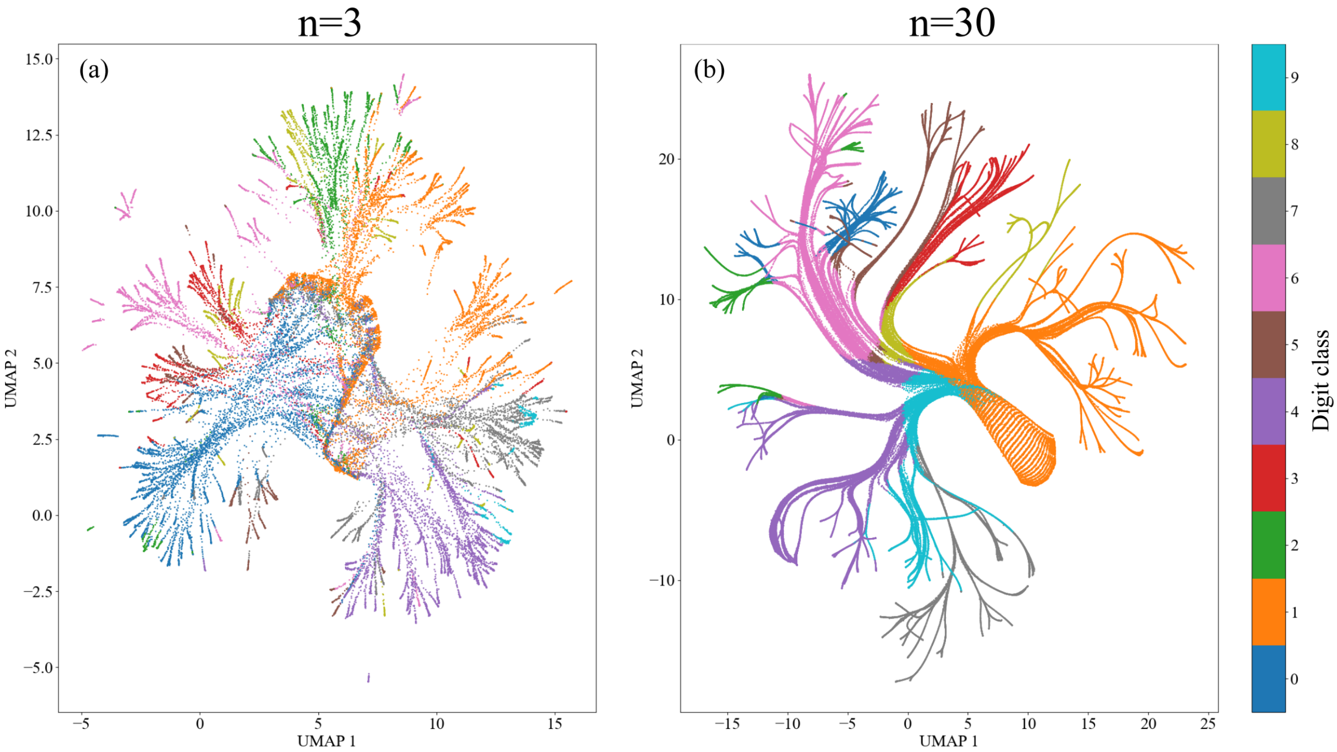

FIG. 4. UMAP embeddings of the memories, for a system with 400 memories, trained on 1000 digits $T_r = 0.85$, other parameters similar to Fig. 1. (a) $n = 3$, for the embedding memories were sampled every 10 epochs, until epoch 1000 when memories stabilize. (b) $n = 30$, for the embedding memories were sampled every 5 epochs, until epoch 2300 when memories stabilize. Colors indicate the highest l^d_μ for any given memory $|M^\mu\rangle$.

first new digit appearing using the example of Figs. 3(e)–3(h), we see that all memories start changing, identically, eventually looking like a mixture of digits, typically 1 and 9 [Fig. 3(g), epoch 900] and embedded as an 8 in the UMAP. Subsequently, a symmetry breaking (or "split") event occurs: While one subset of memories goes back to look like 1, the other subset of memories initially resembles a mixture of 4, 7, 9 (epoch 1250). Thus, the initial mixed state around epoch 900 appears to be a *saddle* of the learning dynamics [25]: Memories first converge toward it in one direction, then upon reaching it, get repelled in another direction, driving the split into two subpopulations. A rigorous demonstration of the existence of saddles in the high n regime of a GHN would require identification and linear classification of its fixed points, which is challenging due to the very high dimensionality. To this end, we also notice that the system spends many epochs close to those mixed states, consistent with the idea that only a few unstable directions exist along which the memories split. For now, we will call these points in memory-space "effective saddles", and will better characterize them for simpler cases below.

New digits are learned when a subpopulation of memories start to resemble mixtures of yet-to-learn digits, before splitting at an effective saddle point to acquire a new digit identity. For instance, the memories in Fig. 3(g) eventually differentiating into 3s and 5s are identical up to epoch 1800, resembling each other at all prior epochs. We also demonstrate the robustness of the dynamics and the location of effective saddles in the presence of noise (see Figs. S.13–S.16 within the SM [28]). It is only towards the very end of learning, when their final identity is fixed, that memories specialize into different prototypes and that more variability appears. All in all, our observations strongly suggest that, despite the high dimensionality of the system, the presence of effective saddles in memory space confer a low-dimensionality and reproducibility to the emergent learning dynamics of GHNs.

Finally, to provide finer-scale resolution of the learning dynamics, we generated UMAP embeddings, by training a larger system (400 memories, 1000 samples) and sampling the memories at fixed epoch intervals, Fig. 4, with a color code indicating the identity of the highest l^μ_d for each memory $|M^\mu\rangle$. This allows us to circumvent the "jumps" in the embedding of Figs. 3(a)–3(d) and to better visualize the branching dynamics during learning. For $n = 3$, we see a complicated, branched network of entangled memories; however, we also see that digits with the same dominating label tend to branch from one another. By contrast, for $n = 30$, a simpler tree structure emerges, consistent with what we have described thus far. We observe a "trunk" of memories with a 1 identity, splitting into different branches with well defined identities, corresponding to the sequence described below (e.g., the leftmost branch correspond to the 4, 7, 9 memories). We also see several sequential splits, for instance the 0 memories are splitting from the 6 memories or the 7 memories are splitting from the 9 memories. Those splits happen sequentially as learning is occurring in later epochs, as can be seen in Movies 3 and 4 within the SM [28]. We also compare visually the tree-like dynamics of different simulations in Fig. S.19 within the

SM [28], largely observing the same topological features and groupings within common subtrees of digits learned together (e.g., 4, 7, 9 vs 3, 5, 8).

D. Three-category systems recapitulate the phenomenology of the dynamics

Motivated by the low-dimensional features of digit learning at high n, we then proceeded to more deeply study simpler versions of the system, hoping to mathematically capture the most salient features of the overall dynamics.

While the UMAP representations introduced in the previous section are useful for a qualitative understanding of the process, there are potential problems with such methods for quantitative analysis [34,35], in particular, the axes of such representations are not interpretable. The $\alpha_{|i\rangle}$s provide a natural coordinate system, but they are difficult to visualize when there are too many samples. However, it turns out that coarse-graining multiple $\alpha_{|i\rangle}$ corresponding to the same digit allows for convenient representations of the dynamics close to saddles. For each memory $|M^{\mu}\rangle$ we thus define aggregated $\bar{\alpha}^{\mu}_{d}$s associated to a digit/category $d \in [0, 9]$ such that

$$\bar{\alpha}^{\mu}_{d} = \sum_{|i\rangle \in \mathcal{T}_d} \alpha^{\mu}_{|i\rangle} \tag{6}$$

where $\mathcal{T}_d$ defines all samples in the training sets labeled with digit/category d.

The fundamental feature of sequential splitting can then be reproduced and visualized for simulations trained with only three categories of digits. In Fig. 5, we represent learning trajectories of all memories for simulations where the training samples only contain 1, 4, 7, using the same embedding as in Fig. 3, for $n = 3, 30$ and $T_r = 1.02$. In the UMAP space, Fig. 5(a) right, the steps in learning are virtually identical to Fig. 3, with a first effective saddle/split localized in a 8 cluster, and a second one in a 9 cluster. Such dynamics are all the more remarkable since neither 8 nor 9 sample digits are included in this reduced training set. Convergence towards those clusters comes from the fact that the effective saddles prior to splitting indeed resembles an 8, then 9. We also more clearly observe in this simplified setting how similar digits are initially identical, then eventually specialize into different prototypes, expanding within one cluster [late epochs in Fig. 5(a)].

In the $\bar{\alpha}_i$ space, Fig. 5(a) left, one observes a first convergence towards an effective saddle, then a split around $\bar{\alpha}_1 = 0.5$, sequentially followed by a second one around $\bar{\alpha}_7 = \bar{\alpha}_4 = 0.5$, indicating that effective saddles are indeed mixtures of digits. In between these two splits, the trajectories are localized along the plane $\bar{\alpha}_1 + \bar{2}\alpha_4 \sim \bar{\alpha}_1 + \bar{2}\alpha_7 \sim 1$. We contrast this behavior with what happens for low n in the feature regime, Fig. 5(b), where the trajectories in the UMAP space do not correspond to clear steps and are not interpretable. In the $\bar{\alpha}_i$ coordinates, Fig. 5(b) left, there is only one global split in a location where all $\bar{\alpha}_7$ and $\bar{\alpha}_4$s are small and the trajectories after the splits are spread out. Such behaviors are generic and do not depend on the digits used: For instance, in Fig. S.20 within the SM [28] we further illustrate what happens for a different set of digits, 1, 7, 9, with very similar properties.

E. Properties of the two-memory system

The splitting of memories at effective saddles thus appears to fundamentally drive the dynamics of learning in the large n regime. We hypothesized and checked that such splitting could be observed for systems with only two memories and two digit categories. Given that the memories all behave identically before the split and in each branch after the split, and that learning, globally, occurs through sequences of splits, we now choose to focus on the study of one single split. To do so, we further reduced the system to two memories and only two input digits to discriminate, which allows for an analytical description of the system, and in particular demonstrates rigorously that the location of the splits are indeed saddles of the dynamics.

Before the split, calling $|A\rangle$, $|B\rangle$ the two vectors corresponding to the two digits to discriminate, the (identical) memories can always be written as a linear combination of the two digits, defining

$$|M\rangle = \alpha_{|A\rangle}\,|A\rangle + \alpha_{|B\rangle}\,|B\rangle \tag{7}$$

for the (common) memory before splitting. In Sec. 3.1.1 and Figs. S.21 and S.22 within the SM [28], we show that the labels are getting quickly correlated so that all labels get to -1 except the two labels corresponding to digits A, B such that $l_A = -l_B$, subsequently called ℓ. The learning dynamics act on the αs and ℓ.

Memories are normalized at each learning epoch such that

$$|\alpha_{|A\rangle}| + |\alpha_{|B\rangle}| \leqslant 1; \tag{8}$$

however, one can demonstrate that the dynamics quickly drive the system to the boundary,

$$|\alpha_{|A\rangle}| + |\alpha_{|B\rangle}| = 1, \tag{9}$$

where the sign of each term in this equation depends on the sign of ℓ (see Sec. 3.1, and Figs. S.21 and S.22 within the SM [28]). In Sec. 3.1.8 [28], we show that all fixed points of the system are in this bidimensional boundary, e.g., we can eliminate $\alpha_{|B\rangle}$ to carry out a nullcline analysis for ℓ and $\alpha_{|A\rangle} = \alpha$. We can write two effective equations for α and ℓ, formally

$$\frac{d\alpha}{dt} = \mathcal{A}(\alpha, \ell), \tag{10}$$

$$\frac{dl}{dt} = \mathcal{L}(\alpha, \ell). \tag{11}$$

The full expressions for functions $\mathcal{A}$, $\mathcal{L}$ are rather complex and given in the SM [28], Eqs. (S.65) and (S.72). Equating them to zero defines two nullclines, respectively called "memory" [$\mathcal{A}(\alpha, \ell) = 0$] and "label" nullclines [$\mathcal{L}(\alpha, \ell) = 0$], see Fig. 6. The dynamics of learning converge towards the 2D curves defined by those nullclines, Fig. 6(a), Movie 5 and Fig. S.23 within the SM [28]. Two-memory systems initially converge towards the points at the intersection of those two nullclines $\mathcal{A}(\alpha, \ell) = \mathcal{L}(\alpha, \ell) = 0$, indicating they are critical points of the full dynamics Fig. 6(b). Importantly, those critical points are the invariant vectors of normalization, i.e., such that $|\Delta M\rangle = \Delta\alpha_{|A\rangle}\,|A\rangle + \Delta\alpha_{|B\rangle}\,|B\rangle$ is proportional to the initial memory $|M\rangle$.

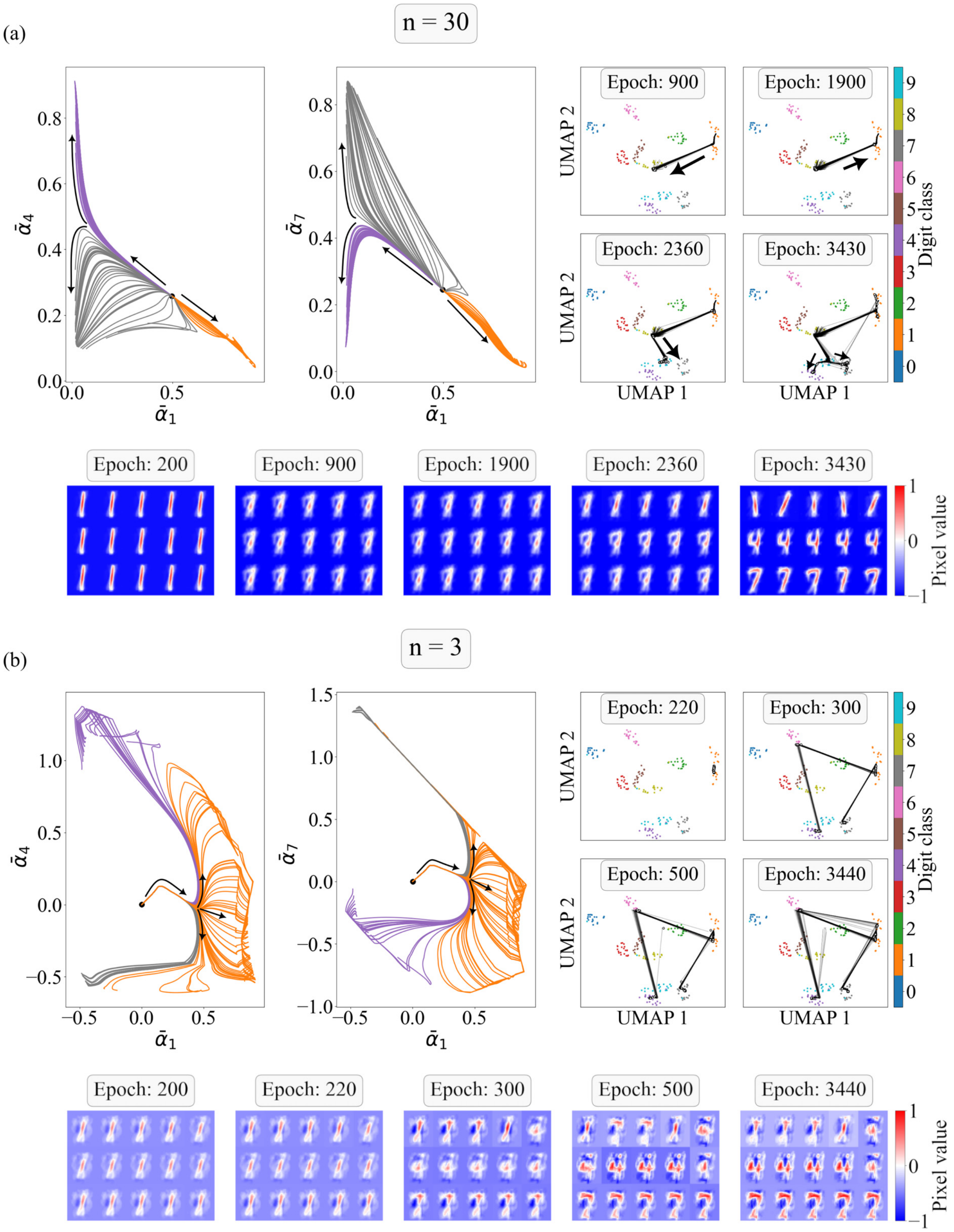

FIG. 5. Splitting dynamics for high and low n. Here the 100-memory system is trained on only three categories - 1, 4, 7 - with 20 training samples per class, and $T_r = 1.02$. We visualize trajectories in the similar UMAP embedding as in Fig. 3 and display samples of the memories on the bottom row. We also display on the left the memory trajectory projected on the α coordinate, computed with the Moore-Penrose pseudo inverse. Trajectories are colored according to the dominating α at the end of the learning (a) $n = 30$. Two successive splits are clearly visible separating 1 from 7, 4 (around epoch 2360 as seen on the UMAP embedding), then 7 from 4. (b) $n = 3$. The splits all happen around the same epoch (300). Notice how there are many more connections between final memories in the UMAP embedding, and how the final memories look like mixtures of digits.

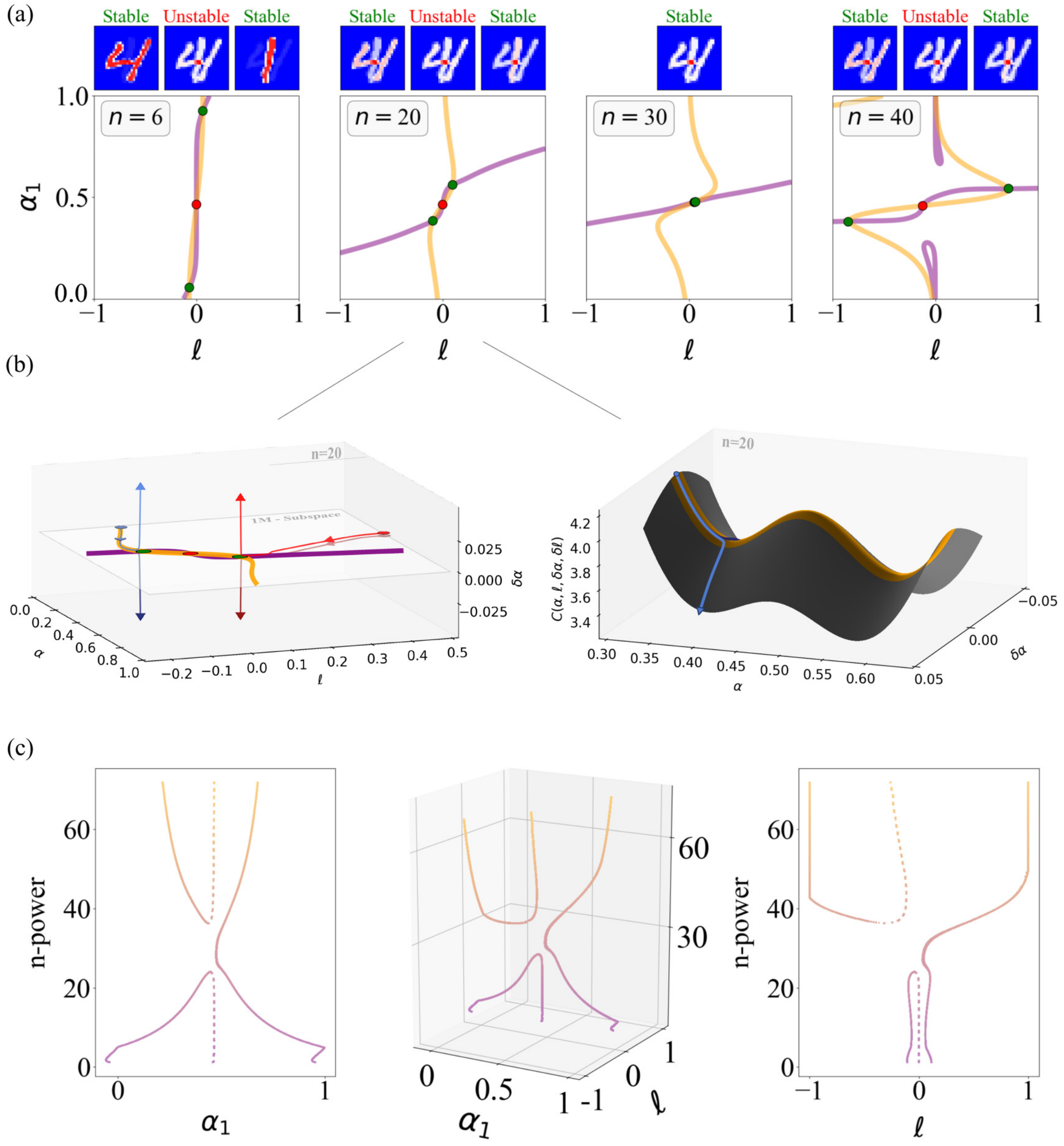

FIG. 6. Bifurcation and dynamics for the two-memory system. In (a), the fixed points as a function of n are illustrated with the nullclines (memory: purple, label: orange). Note that in the presence of 3 fixed points, the "center" point is unstable. (b) Examples of the splitting dynamics for the two-memory system for $n = 20$ in higher dimension. On the left, one displays the dynamics of two pairs of memories in the α, $\delta\alpha$, ℓ space, in shades of red and blue. Initial conditions are indicated by circles, blue and red arrows on the trajectories indicate the direction of the dynamics. Nullclines corresponding to (a) are also indicated. We see how the memories first converge to the nullclines, before splitting at the saddles in the $\delta\alpha$ direction. On the right, we show the corresponding cost functions as a function of α, $\delta\alpha$, while ℓ and $\delta\ell$ are parameterized by their α counterparts; $\ell(\alpha)$ is defined using the label nullcline, while $\delta\ell(\delta\alpha)$ is defined using the typical splitting trajectory (see Sec. 7.5 within the SM [28]). We clearly see the saddle shapes close to the green fixed points of (a). In (c), we show the bifurcation diagram of a two-memory system with two identical memories, using n as a control parameter, and α and ℓ to show the fixed points. The center plot of (a) is the complete 3D bifurcation diagram of this system, the left and right plots show projections on planes of constant α and ℓ respectively. All simulations in this figure are for a two-memory system with $T_r = 0.89$, for training data with two sample digits 1, 4 corresponding to $|A\rangle$, $|B\rangle$ with $\langle A\,|A\rangle = 753$, $\langle A|\,B\rangle = 494$ and $\langle B|\,B\rangle = 719$.

F. Bifurcation diagram define three different regimes for critical points

The two nullclines are sigmoidal, and one observes three qualitatively different regimes going from bistability to monostability to again bistability as n is increased, see e.g., Fig. 6(c) using digits 1, 4 as examples. Multistability in this context means there can be multiple critical points for the learning dynamics. We will label those critical points based on the stability of the system where the two memories are forced to be identical, i.e., the dynamics driven by the memory and label nullclines Eqs. (10) and (11). Stable fixed points for the system Eqs. (10) and (11) are saddles for the full dynamics, since the dynamics are initially driven to those critical points, while the unstable fixed points are not saddles because they are not visited by the full dynamics. When there is more than one critical point, the location of the saddles thus depends on the initial conditions of the system. The critical points themselves appear like digits for low n (first bistable region), then like a mixture of digits in the monostable phase for intermediate n, and again gradually look like single digits in the second bistable region as n is further increased.

Figure 6(a) illustrates nullclines for various values of n. For low n, both nullclines are almost vertical, parallel to the α axis, defining a critical valley at a fixed value of ℓ, but where future splits between memories are very sensitive to noise and initial conditions. This is consistent with the stochasticity observed in the full system (see Movie 1 within the SM [28]). Intuitively, vertical nullclines (close to $\ell = 0$) mean that the system is unable to label the internal memories properly: all labels stay close to 0 [as can be seen on the bifurcation diagram, Fig. 6(a) right panel, low n] and there are only weak biases in one direction or the other depending on the fixed point.

As n is increased Fig. 6(a), $n = 20$, the slope of the memory nullcline decreases first, so the system has three critical points. Interestingly, because the label nullcline stays relatively vertical, they correspond to relatively high and low values for α parameters, meaning that the critical points look very much like one of the initial digits but with some background of the other one. Notice, however, that the ℓ stay low, meaning that while the memories are relatively well defined, their labels are not.

As n is further increased, Fig. 6(a), $n = 30$, the stable critical points disappear through a saddle-node bifurcation (see Fig. S.25 within the SM [28]), and remarkably, only the fixed point close to $\alpha = 0.5$, $\ell = 0$ survives. Intuitively, the system gets more selective in its recognition of proper digits, so that any mixture of digits can (and should) not be categorized as either category. The perfect mixture gets an ambiguous $\ell = 0$ label, which is the only fixed point surviving. In Fig. S.26 within the SM [28], we show that the system is close to a pitchfork bifurcation, due to the fact that most pixels in the initial pictures take values close to 1 or -1, so that $\langle A|A\rangle \sim \langle B|B\rangle$.

As n is further increased, a new saddle-node bifurcation occurs, where ℓ values initially become significantly higher than 0. This comes from the fact that the memory nullclines become increasingly horizontal, close to $\alpha = 0.5$. This is the more intuitive regime in the generalized Hopfield energy landscape: as n is increased, the "energy" of the saddle point gets very frustrated between the two digits, so that the only critical point for the one memory system should be exactly a superposition of both digits corresponding to $\alpha \sim 0.5$. However, there is a slight bias in the loss function due to the label, so one gets two stable fixed points, in which one digit slightly dominates the other in the memory, but with an unambiguous categorical label ℓ. As n is further increased, those biases on the labels allow for further symmetry breaking between the two fixed points, which eventually look increasingly like "pure" digits. See Fig. S.27 within the SM [28] for additional n values.

G. Saddle dynamics

The full two-memory system is four-dimensional, with two alpha coordinates and two label coordinates. As said above and illustrated in Figs. S.21 and S.22 within the SM [28], the system quickly reduces to two dimensions α, ℓ, but to study the full dynamics we can also define two deviations $\delta\alpha$, $\delta\ell$, initially extremely small and symmetrical {see Eqs. (S.85)–(S.87) and Eq. (S.88) within the SM [28]}. Examples of the dynamics of two memories are shown in Fig. 6(b) in the α, ℓ, $\delta\alpha$ space. As mentioned earlier, after a transient phase, $\delta\alpha$ is essentially 0, meaning that the dynamics are initially structured by the α, ℓ nullclines, Eqs. (10) and (11). Both memories converge to the (closest) critical point defined by the α, ℓ nullclines. There, they split, so that both $\delta\alpha$, $\delta\ell$ rapidly increase Fig. 6(b), left. In Fig. 6(b), right, we represent the corresponding cost functions as a function of α, $\delta\alpha$ and examples of trajectories followed. We see that the landscape is shaped as a saddle, thus confirming that the critical points are saddle points, and explaining the corresponding split.

In Figs. S.29 and S.30 within the SM [28], we further compare the simulated trajectories with analytical computations of the small deviations $\delta\alpha$, $\delta\ell$ close to the saddle, with excellent agreement.

H. Feature-to-prototype transition for the two-memory system

While the bifurcation diagram described in Fig. 6(c) above depends on n, it relates to saddles in the learning dynamics, but does not inform on the feature-to-prototype transitions, which relates to the fixed points. Such fixed points can, however, be computed analytically for the two-digit/two-memory system. Figure 7 illustrates final memories for different values of n, T_r for two cases: one where digits are of the same category (so having the same ℓ), Figs. 7(a), 7(c), and 7(e), and the other where digits are of different categories (so opposite ℓ), Figs. 7(b), 7(d), and 7(e). We illustrate how the αs of the fixed points are changing as a function of n, T_r for both cases. Clearly, as both parameters are increased, the final states change from a mixture of both digits (looking like features), to a more defined digit (looking like a prototype). Consistently, one goes from fixed points with one positive and one negative α, to fixed points where only one α dominates [from top left to bottom right in Figs. 7(c), 7(e), and Figs. 7(d), 7(f)].

It is also possible to analytically compute the transition line where a single digit yields the same cost as a perfect mixture of digits, presumably corresponding to the feature-to-prototype transition (see Sec. 6.1 within the SM [28]). We compare it to actual simulations, with excellent agreement [white line in Figs. 7(e) and 7(f)].

For the intradigit classification (i.e., same category), one gets pure prototypes for high n, T_r, in the sense that the α contribution from one of the digits completely vanishes. Also, there is a clear threshold in n from which one α suddenly increases above 0.5, thus defining two regimes, and reminiscent of the behavior of the maximum of α displayed in Fig. 2(a). However, for the interdigit discrimination (i.e., different

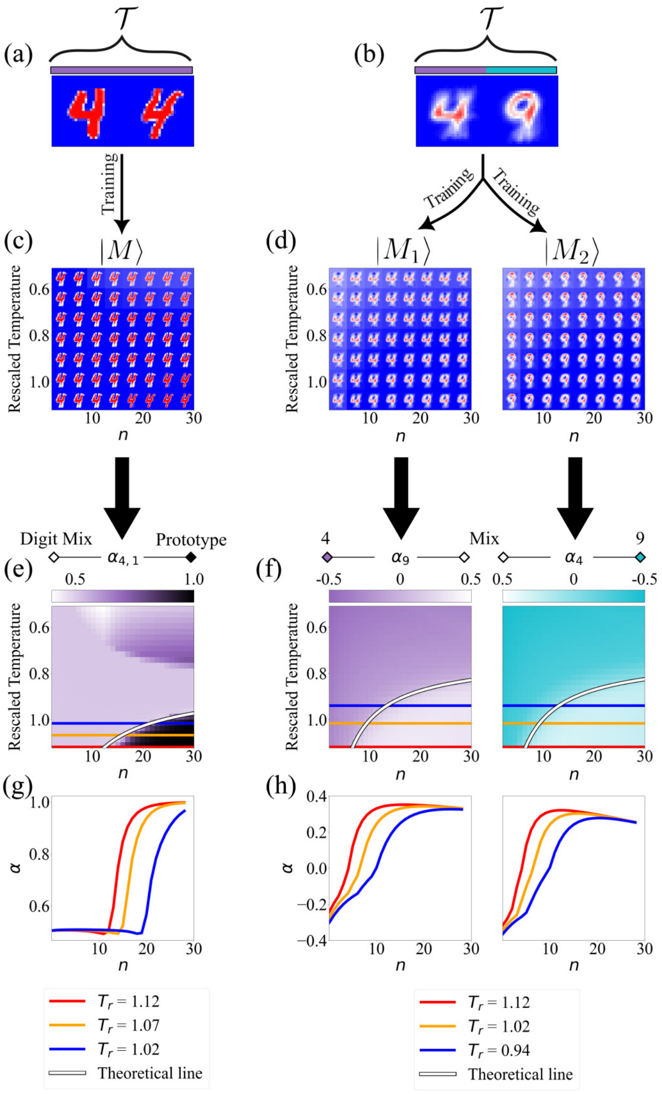

FIG. 7. Two-memory system recapitulates a feature-to-prototype transition. (a) Training set used to study the one-memory intradigit system; (b) the training set for the two-memory inter-digit system. (c), (e), (g) Intradigit specification. We show the final state of the memory as a function of n, T_r (c), the corresponding α_4 (e), and curves of α_4 as a function of n for three high temperatures (g). The white line on (e) indicates when the score of a perfectly mixed digit is the same as the score of either digit in (a), coinciding with the feature-to-prototype transition. (d), (f), (h) Interdigit specification. We show the final state of memory 1 (left) and 2 (right) as a function of n, T_r (d), the corresponding α_9 of memory 1 (left) and α_4 of memory 2 (right) (f), and curves of α_9 (left) and α_4 (right) as a function of n for 3 high temperatures (h). The white line on (f) indicates when the score of a perfectly mixed digit is the same as the score of either digit in (b), coinciding with the feature-to-prototype transition.

categories), the behavior of both αs is smooth and we do not see a clear separation between two regimes. Furthermore, even for big n, T_r, one still sees a small but nonzero contribution of both digits in the fixed point. So one never gets to a "pure" digit representation as expected from a true prototype: rather the fixed points look like some mixed saddles in the high and low n limit in Fig. 5.

We thus conclude that two-memory systems recapitulate some of the salient aspects of the feature-to-prototype transition observed in the full system, suggesting it is not an aspect of the learning that is due to the high number of memories in the system but that, surprisingly, the different "phases" in α are more visible for intradigit categories than for interdigit categories. This is, however, consistent with the observation made on the full system that initially similar memories differentiate into prototypes late in the training : it is likely only when two memories have the same label that the intra-digit split is happening, placing almost all αs to 0 and thus defining a proper prototype.

I. Learning and final memory statistics in the expanded system are structured by bifurcations

We have thus identified two aspects of the two-memory system: a bifurcation diagram structuring the saddles visited during learning, and a feature-to-prototype transition on the final states of the two-memory system. Importantly, while those two aspects are correlated (because they both depend on n, T_r), they are independent in the sense that one does not need the knowledge of the bifurcation diagram of the saddles to compute the final states of the memories. We know that the feature-to-prototype transition generalizes from systems with few memories to bigger ones, but it is not clear if the number or nature of saddles matter in any way for bigger systems. We thus now re-expand the number of memories and samples to study how/if these properties exhibited for the two-memory system generalize.

Because of the combinatorial explosions of possible memories and saddles, we move back to numerical explorations to study the influence of saddles on learning. We first restrict ourselves to a system with 100 memories, three categories (1, 7, 8) and 20 samples per category. To explore different learning trajectories, possibly corresponding to different saddles, we initialized the memories close to different digits, then performed learning for different temperatures Fig. 8(a)–8(c) and focus on the 1 vs 7 discrimination. We first recovered multiple saddles (see Sec. 3.2 within the SM [28]) in learning: we observe again at least three regimes for high temperature $T_r \sim 1$, with at least two bistability regions for low and high n, and an intermediate region where the system converges towards a single saddle.

Thus, a similar bifurcation diagram to the two-memory case is also observed for systems with many more memories. Importantly, we also observe that the *proportion* of final memories of a given identity depends on initial conditions, and thus on the saddle first visited Figs. 8(a)–8(c), a property that can not be observed on systems with two memories since each memory stabilizes to either fixed point. Consistent with this, the proportion of 1 and 7 in the final memories is correlated to the respective values of α_1 and α_7 at the saddle visited. For instance, for $T_r \sim 1$ when there are more saddles (small/big n), when the system is initialized close to 1 (resp 7), the proportion of 1 (resp 7) in the final memories is very high for most parameters, while it is very low if the system is initialized close to 7 (resp 1), Fig. 8(c). Only for intermediate n, when only one saddle is visited irrespective of the initial conditions, do we observe a balance between 1s and 7s in the

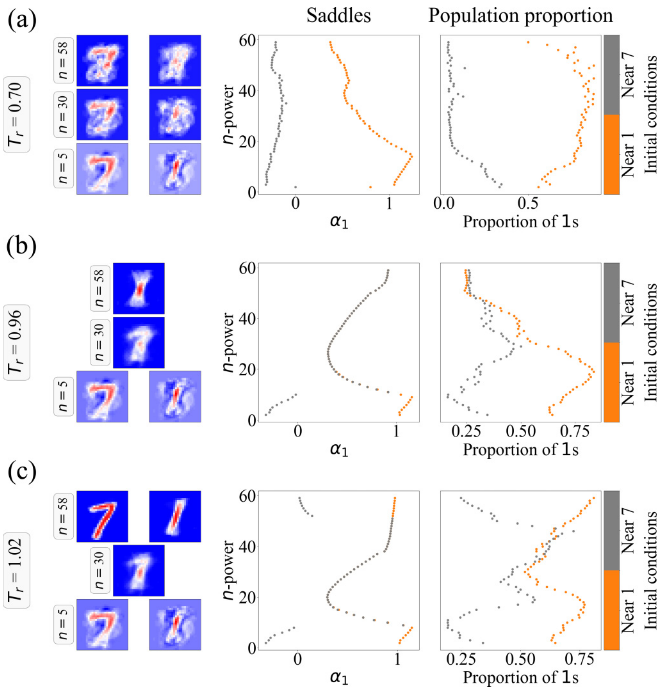

FIG. 8. Bifurcation of saddles correlates to changes of proportions. We show results of simulations respectively initialized close to a 1 sample digit (orange dots) and close to a 7 sample digit (grey dots). From left to right, we show examples of saddles reached in the left panel, α_1 coordinates of the saddles in the middle panel, and the relative proportion of 1 vs 7 in the right panel as a function of n. (a) $T_r = 0.7$. (b) $T_r = 0.96$. (c) $T_r = 1.02$. All networks contain 100-memories, and are trained on the same training set of 60 digits (evenly composed of 1s, 7s and 8s) with a training rate of 0.01.

final memories, when the (unique) saddle is a mixture of 1 and 7. The final state reached after learning thus appears path dependent, and the number and nature of saddles are relevant for the final statistics of the memories learned.

Importantly, we also see a clear T_r dependency on both the bifurcation diagram and the proportion of memories, Figs. 8(a)–8(c). We examined what happens for the full system as we vary T_r, in a range of n where we expect few saddles $\{$typically $n \in [20, 30]$ according to Figs. 8(b) and 8(c)$\}$. In Fig. 9, we show samples and the number of memories for each category after training, as a function of temperature (using trained labels as a proxy, see Movies 6–8 within the SM [28]). For lower $T_r \lesssim 0.85$, all digits are represented in the memories. However, as T_r increases, some memories disappear and instead start looking like the typical first saddle observed during training Fig. 9(a). Remarkably, as T_r is increased, there is a clear order in the disappearance of memories: first 0 and 2, then 3, 8; 5, 6, and 4, 7, 9, leaving only 1 in the end when $T_r > 1$. This order essentially is the inverted order of digits learned during one instance of successful training of the system, compare, e.g., Fig. 3(h). This suggests that as T_r is increased, the system is no longer able to split in some directions. Instead, it remains stuck at saddles close to the root of the tree displayed in Fig. 3(e) as T_r is increased.

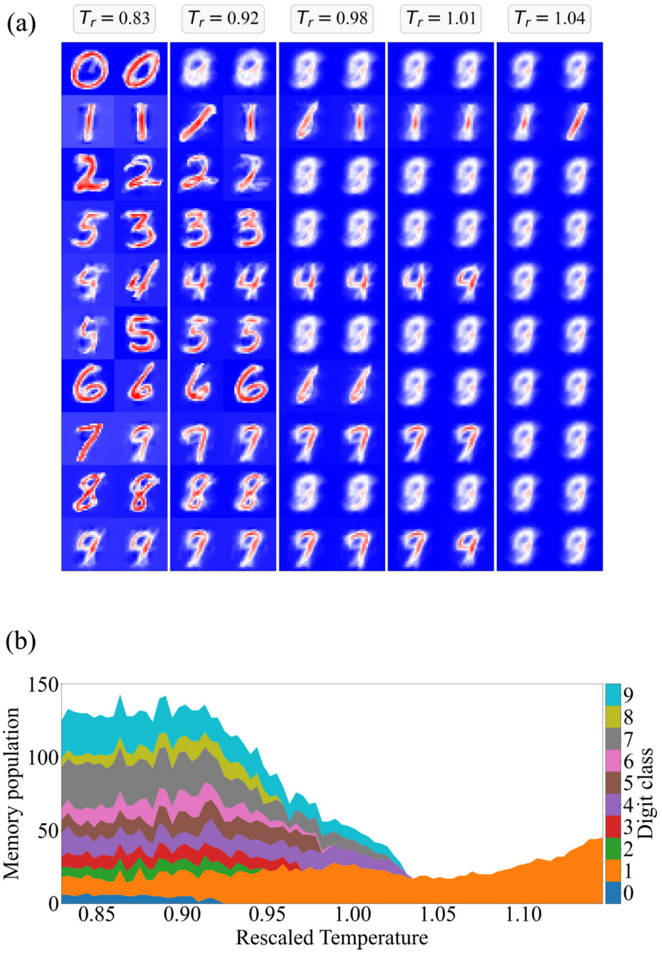

FIG. 9. Training for varying temperature for $n = 25$. As the temperature increases, some categories are no longer properly recognized. (a) Samples of final memories for different temperatures. (b) Population for each memory (as inferred from their label) as a function of Temperature after training. The network size was of 100 memories, with gaussian random initial conditions, and a training rate of 0.005. The training set used was the same as in Fig. 1, containing 200 digits (20 of each class).

III. DISCUSSION

Artificial neural networks offer examples of nontrivial self-organizing dynamics, and as such constitute easy-to-simulate comparison points to real-life complex networks. In this paper, we studied the learning dynamics of generalized Hopfield networks trained to classify MNIST, previously known to exist in two broadly defined regimes (feature and prototype based) as a function of hyper parameters characterizing the non-linearity of the energy landscape.

We visualized and studied the learning dynamics of such networks. We especially focused on the prototype dynamics (for higher n) where learning is most low-dimensional and reproducible: Memories localize first at saddles, corresponding to mixtures of digits. Subsequently memories split, before reaching either a new saddle or specializing into an actual digit. Memories with the same eventual digit identity largely follow the same path in memory space during learning and only specialize into different prototypes in terminal epochs of learning. The order in which memories are learned is reproducible across simulations for large n.

We explore the dynamics of this complex learning scheme through simpler versions of the systems with fewer memories, establishing the local low dimensionality of the processes, and allowing us to better characterize the most salient properties of the learning dynamics. We reveal that the number and nature of possible saddles of the dynamics depends on n, T_r, in particular, we see for intermediate n and higher temperature a regime where there are a smaller number of saddles in the dynamics of learning, which themselves appear and disappear via saddle-node bifurcations at both lower and higher n, respectively.

Our results suggest the following view of the learning dynamics in the prototype regime: for lower n, T_r, there are multiple saddles, and thus as shown in Fig. 3, we generically observe more path dependency and randomness in learning, leading to different proportions of memories with a given digit identity. However, for mid range n (typically $n \in [20, 30]$ for $T_r \sim 0.89$), the system is in a regime where fewer saddles exist, most likely due to the first saddle-node bifurcation [Fig. 6(b), $n \sim 30$]. This is consistent with the plateau observed in Fig. 2(a) right for higher temperatures: Memories are "trapped" at various saddles, as visible in the UMAP plot in Fig. 2(b) where memories do not distribute well within one digit cluster. It is then only for even bigger n ($n > 30$) that more saddles appear again. Importantly, these saddles correspond to mixtures of digits, but "biased" towards one digit. This allows the learning to have more path dependency, but locally (i.e., within a digit category), giving rise to many more prototypes as final states. This provides a rationale as to why the memories in the UMAP plot in Fig. 2(b) are spread out for larger n. Manifestly, the location and nature of saddles influence the proportion of final memories in a given digit category. Interestingly, as we increase T_r, the system gets increasingly stuck in saddles, following the pathway of learning. This is consistent with the observation that the learning dynamics of a digit category is low dimensional, with the detrimental effect that it is also "easy" to sequentially block those directions during learning. This does suggest that there is a "sweet spot" to ensure a desirable prototype distribution over all digits: A low number of saddles between digit categories is needed to have a good number of memories corresponding to all digits, but if the temperature is too high learning can get stuck at those saddles (which then become stable fixed points).

We wish to highlight that the feature-to-prototype transition is not restricted to systems with a large number of memories and, importantly, the pure prototype regime where only a single digit (α) dominates appears to come from intradigit classification, rather than interdigit classification. This is consistent with the fact that memories with the same eventual label largely appear to change together, as can be seen in Fig. 3(g) and Movies 1 and 2 within the SM [28]. It is only when memories have the same final label that intradigit interaction ensues and that they become pure prototypes.

How general are our results? Of note, generalized Hopfield networks have been related to both diffusion models [22] and transformers [2], the architecture at the core of the success of current Large Language Models, such as chatGPT. In particular, Ramsauer *et al.* [2] use an explicit energy with an exponential for the function f defined in Eq. (2), thus do not have the equivalent of a hyperparameter n, but include an inverse temperature β (which, as seen here, can have similar effects as n). They did not study the dynamics of learning and instead focused on the types of attractors, observing both single patterns (corresponding to prototypes) and mixtures of "close" patterns, which within our framework would likely correspond to saddles. However, they did not explore the anticipated bifurcations. It is unclear if there is any way those saddles would split into well-defined prototypes like in our case. Interestingly, they relate their dynamical updates to self-attention in transformers, which linearly combines input samples and would thus correspond to changes in α, in our context. They also observed the distribution of activation patterns in layers of actual transformers (similar to distribution of αs), and observe changes reminiscent of what we observed here when n is increased, e.g., with some mid-late layers activated by very few patterns, while many patterns broadly activate early layers. This suggests that a feature-to-prototype transition might be a generic property happening within different layers of neural networks, which provides further motivation for case studies of more tractable systems as we have explored here.

The tension between the high dimensions of the sample space and low dimension of the dynamics is very reminiscent of what happens in complex (biological) systems, a primary motivation for the present study. In the prototype regime, memories with the same terminal identity differentiate sequentially, moving from one saddle to the next, before splitting. This resembles the classical model of landscape exploration during the time course of a differentiating cell proposed by Waddington. We summarize Waddington's qualitative view with four properties:

(1) valleys can "split", corresponding to binary decisions,

(2) those splits are localized, corresponding to well-defined points in the possible space of cellular states,

(3) between splits, the dynamics are robust or "canalized" along the valleys,

(4) the overall dynamics are self-organized, and modulated by slowly changing underlying weights of unknown nature.

The learning dynamics in the prototype regime present similar properties, that we more precisely quantified. As we establish here, internal memories split (property 1) at well-defined saddles (property 2). Numerical and analytical calculations (two-memory section, Fig. 6) reveal that the trajectories of identical memories between splits are attracted to one-dimensional manifolds (property 3). The dynamics of memories are also self-organized, congruent with the dynamics of the labels (property 4).

Waddington's landscape's ideas also lead to biological predictions, that we can relate to the learning dynamics we see. For instance, a nontrivial consequence of property 2 is the existence of a well-defined hierarchy of states, with intermediate progenitor or stem fates. Indeed, one can experimentally maintain cells in stem fates with exposure to proper morphogens [17], thus suggesting 1. they are indeed saddles of the differentiation dynamics, and 2. the few corresponding unstable directions can be stabilized through external means.

We observed the exact same properties with the tuning of hyperparameters n, T_r : the system precisely stays stuck in the saddles normally visited during learning (Figs. 2 and 9), at different levels of the hierarchy depending on the values of T_r. All in all, the congruence of the aforementioned properties lead us to propose a useful analogy, if not equivalence, between prototype learning and Waddingtonian dynamics for cellular differentiation.

What are the implications of this connection? On the theoretical side, our results are consistent with the idea that learning dynamics are structured by a sequence of hierarchically organized saddles, a problem recently formalized for cellular differentiation [14]. Extending such theory could help in understanding more broadly how both machine learning and differentiation work. In particular, it is not clear how self-organization occurs in both cases. Are the saddles of the dynamics fixed points, with a slow variation of self-organized control parameters that drive bifurcations leading to splits [36]? Coming back to biology, Matsushita *et al.* [10,11] have proposed a model for self-organized cell differentiation where interacting transcriptional states slowly regulate epigenetic states. They observed that, for random networks, progenitor states display oscillations before "localizing" their dynamics towards multiple steady states, through bifurcations controlled by the slow varying epigenetic states. This two-layer organization, with induced bifurcations, is very reminiscent of the two-level architecture studied here with memories and labels, and also consistent with Waddington's view of separating the landscapes from weights controlling them. Other aspects of the feature-to-prototype transition, or the multiple regimes of learning that we observe, might also apply to differentiation biology. One could imagine that more random routes to (de)differentiation might exist, e.g., in a context where chromatin states are less well defined, possibly corresponding to the low n regime, with the appearance of spurious attractors [18,19].

Hopfield networks offer robust ways to encode discrete states [24], and have been already used to model the multiplicity of cellular states and their transitions [18,19,37]. Connecting such frameworks with our simulations, one natural question is if/how the dynamics of pattern selection or transition can relate to the dynamics of learning those patterns. In biology, this is a well-defined question relating differentiation or reprogramming routes observed today to the evolution of those cellular states, which we know little about [38]. As described here, the motions in generalized Hopfield landscapes during learning are reproducible and locally low dimensional, which fits Waddington's picture for differentiation. The simplest explanation of such similarity between both dynamics would be that differentiation represents an instance of ontogeny (the trajectory within the space of cellular states) reflecting phylogeny (the evolutionary pathways leading to those states). For some cell types, there is evidence of such concordance [39]. More generally, alignment between the directions of evolution and cellular dynamics have been recently observed in multiple contexts [21,36, 40–42]. In this view, Waddingtonian landscape dynamics observed today for differentiation could be a spandrel [43] of an evolutionary dynamics analogous to the Waddingtonian learning that we see here. Evolutionary statements of those sorts are extremely challenging to prove experimentally. Quantifying the emergence of hierarchical landscapes in tractable models of complex systems might thus provide indications of such universal aspects across fields from machine learning to biology.

ACKNOWLEDGMENTS

We thank members of the François and Mani groups for useful discussions. P.F. is supported by the Natural Sciences and Engineering Research Council of Canada, Discovery Grant Program, the Canadian Institutes of Health Research, Project Grant Program, and the Fonds Courtois. M.M. was supported by The National Science Foundation-Simons Center for Quantitative Biology at Northwestern University and the Simons Foundation Grant No. 597491. M.M. is a Simons Investigator. This project has been made possible in part by Grant No. DAF2023-329587 from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation.

[1] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA **79**, 2554 (1982).

[2] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff *et al.*, Hopfield networks is all you need, arXiv:2008.02217.

[3] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, in *Advances in Neural Information Processing Systems*, Vol. 4 (NIPS 2014), pp. 2933-2941.

[4] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, The loss surfaces of multilayer networks, in *Artificial Intelligence and Statistics* (PMLR, 2015), pp. 192–204.

[5] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, Qualitatively characterizing neural network optimization problems, arXiv:1412.6544.

[6] C. Paquette, K. Lee, and F. Pedregosa, and E. Paquette, SGD in the large: Average-case analysis, asymptotics, and stepsize criticality, in *Proceedings of Thirty Fourth Conference on Learning Theory* (2021), Vol. 134, pp. 3548–3626.

[7] J. W. Rocks and P. Mehta, Memorizing without overfitting: Bias, variance, and interpolation in overparameterized models, Phys. Rev. Res. **4**, 013201 (2022).

[8] R. Shwartz-Ziv and N. Tishby, Opening the black box of deep neural networks via information, arXiv:1703.00810.

[9] J. Mao, I. Griniasty, H. K. Teoh, R. Ramesh, R. Yang, M. K. Transtrum, J. P. Sethna, and P. Chaudhari, The training process of many deep networks explores the same low-dimensional manifold, Proc. Natl. Acad. Sci. USA **121**, e2310002121 (2023).

[10] Y. Matsushita, T. S. Hatakeyama, and K. Kaneko, Dynamical systems theory of cellular reprogramming, Phys. Rev. Res. **4**, L022008 (2022).

[11] Y. Matsushita and K. Kaneko, Generic optimization by fast chaotic exploration and slow feedback fixation, Phys. Rev. Res. **5**, 023017 (2023).

[12] S. Liu, P. Lemaire, E. Munro, and M. Mani, A mathematical theory for the mechanics of three-dimensional cellular aggregates reveals the mechanical atlas for Ascidian embryogenesis, bioRxiv (2022), doi:10.1101/2022.11.05.515310.

[13] D. Thura, J. Cabana, A. Feghaly, and P. Cisek, Integrated neural dynamics of sensorimotor decisions and actions, PLoS Biol. **20**, e3001861 (2022).

[14] D. Rand, A. Raju, M. Sáez, F. Corson, and E. Siggia, Geometry of gene regulatory dynamics., Proc. Natl. Acad. Sci. USA **118**, e2109729118 (2021).

[15] C. H. Waddington, *The Strategy of the Genes* (Routledge, New York, 1957), p. 270.

[16] V. Alba, J. E. Carthew, R. W. Carthew, and M. Mani, Global constraints within the developmental program of the *Drosophila* wing, eLife **10**, e66750 (2021).

[17] T. Graf and T. Enver, Forcing cells to change lineages, Nature (London) **462**, 587 (2009).

[18] A. Lang, H. Li, J. Collins, and P. Mehta, Epigenetic landscapes explain partially reprogrammed cells and identify key reprogramming genes, PLoS Comput. Biol. **10**, e1003734 (2014).

[19] S. T. Pusuluri, A. H. Lang, P. Mehta, and H. E. Castillo, Cellular reprogramming dynamics follow a simple 1D reaction coordinate, Phys. Biol. **15**, 016001 (2018).

[20] M. Sáez, R. Blassberg, E. Camacho-Aguilar, E. Siggia, D. Rand, and J. Briscoe, Statistically derived geometrical landscapes capture principles of decision-making dynamics during cell fate transitions, Cell Syst. **13**, 12 (2022).

[21] K. Husain and A. Murugan, Physical constraints on epistasis, Mol. Biol. Evol. **37**, 2865 (2020).

[22] B. Hoover, H. Strobelt, D. Krotov, J. Hoffman, Z. Kira, and D. H. Chau, Memory in plain sight: A survey of the uncanny resemblances between diffusion models and associative memories, arXiv:2309.16750.

[23] D. Krotov and J. J. Hopfield, Dense associative memory for pattern recognition, Advances in Neural Information Processing Systems **29**, 1172 (2016).

[24] D. Krotov and J. J. Hopfield, Dense associative memory is robust to adversarial inputs, Neural Comput. **30**, 3151 (2018).

[25] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering* (CRC Press, Boca Raton, FL, 2018).

[26] N. Goldenfeld and C. Woese, Life is physics: Evolution as a collective phenomenon far from equilibrium, Annu. Rev. Condens. Matter Phys. **2**, 375 (2011).

[27] M. Demircigil, J. Heusel, M. Löwe, S. Upgang, and F. Vermet, On a model of associative memory with huge storage capacity, J. Stat. Phys. **168**, 288 (2017).

[28] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevResearch.6.033098 for complementary calculations, figures and movies.

[29] Y. LeCun, The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/ (1998).

[30] https://github.com/nacer-eb/KrotovHopfieldWaddington

[31] S. L. Freedman, B. Xu, S. Goyal, and M. Mani, A dynamical systems treatment of transcriptomic trajectories in hematopoiesis, Development **150**, dev201280 (2023).

[32] M. Yampolskaya, M. Herriges, L. Ikonomou, D. Kotton, and P. Mehta, scTOP: Physics-inspired order parameters for cellular identification and visualization, Development **150**, dev201873 (2023).

[33] L. McInnes, J. Healy, and J. Melville, UMAP: Uniform manifold approximation and projection for dimension reduction, arXiv:1802.03426.

[34] E. M. Johnson, W. Kath, and M. Mani, EMBEDR: Distinguishing signal from noise in single-cell omics data, Patterns **3**, 100443 (2022).

[35] T. Chari and L. Pachter, The specious art of single-cell genomics, PLoS Comput. Biol. **19**, e1011288 (2023).

[36] P. François and V. Mochulska, Waves, patterns and bifurcations: A tutorial review on the vertebrate segmentation clock, Phys. Rep. **1080**, 1 (2024).

[37] O. Karin, EnhancerNet: A model for enhancer selection in dense regulatory networks captures the dynamics of cell type specification, bioRxiv (2024), doi:10.1101/2024.02.03.578744.

[38] D. Arendt, J. M. Musser, C. V. Baker, A. Bergman, C. Cepko, D. H. Erwin, M. Pavlicev, G. Schlosser, S. Widder, M. D. Laubichler *et al.*, The origin and evolution of cell types, Nat. Rev. Genet. **17**, 744 (2016).

[39] D. Arendt, The evolution of cell types in animals: Emerging principles from molecular studies, Nat. Rev. Genet. **9**, 868 (2008).

[40] C. Furusawa and K. Kaneko, Formation of dominant mode by evolution in biological systems, Phys. Rev. E **97**, 042410 (2018).

[41] T. U. Sato, C. Furusawa, and K. Kaneko, Prediction of cross-fitness for adaptive evolution to different environmental conditions: Consequence of phenotypic dimensional reduction, Phys. Rev. Res. **5**, 043222 (2023).

[42] K. Kaneko, Constructing universal phenomenology for biological cellular systems: An idiosyncratic review on evolutionary dimensional reduction, J. Stat. Mech. (2024) 024002.

[43] S. J. Gould and R. C. Lewontin, The spandrels of San Marco and the Panglossian paradigm: A critique of the adaptationist programme., Proc. R. Soc. Lond. B **205**, 581 (1979).