Representational drift as a result of implicit regularization

Aviv Ratzon[1,2*], Dori Derdikman[1], and Omri Barak[1,2]

[1] *Rappaport Faculty of Medicine, Technion - Israel Institute of Technology, Haifa 31096, Israel*

[2] *Network Biology Research Laboratory, Technion - Israel Institute of Technology, Haifa 32000, Israel,*

*Aviv.Ratzon@campus.technion.ac.il

Recent studies show that, even in constant environments, the tuning of single neurons changes over time in a variety of brain regions. This representational drift has been suggested to be a consequence of continuous learning under noise, but its properties are still not fully understood. To uncover the underlying mechanism, we trained an artificial network to perform a predictive coding task. After the loss converged, the activity slowly became sparser. We verified the generality of this phenomenon across modeling choices. This sparseness is a manifestation of drift in the solution space to a flatter area. It is consistent with recent experimental results demonstrating that CA1 spatial code becomes sparser after familiarity. We conclude that learning is divided into three overlapping phases: Fast familiarity with the environment, slow implicit regularization, and a steady state of null drift. These findings open the possibility of inferring learning algorithms from observations of drift statistics.

What do we mean when we say that the brain represents the external world? One interpretation is the existence of neurons whose activity is tuned to world variables. Such neurons have been observed in many contexts: place cells [1, 2] – which are tuned to position in a specific context, visual cells [3] – which are tuned to specific visual cues, neurons that are tuned to the execution of actions [4] and more. This tight link between the external world and neural activity might suggest that, in the absence of

environmental or behavioral changes, neural activity is constant. In contrast, recent studies show that, even in constant environments, the tuning of single neurons to outside world variables gradually changes over time in a variety of brain regions, even long after good representations of the stimuli were achieved. This phenomenon has been termed *representational drift*, and has changed the way we think about the stability of memory and perception, but its driving forces and properties are still unknown [5, 6, 7, 8, 9] (see [10, 11] for an alternative account).

There are at least two immediate theoretical questions arising from the observation of drift – why does it happen, and whether and how behavior is resistant to it [12, 13]? One mechanistic explanation is that the underlying anatomical substrates are themselves undergoing constant change, such that drift is a direct manifestation of this structural morphing [14]. A normative interpretation posits that drift is a solution to a computational demand, such as temporal encoding [15], 'drop-out' regularization [16], exploration of the solution space [17], or re-encoding during continual learning [12]. Several studies also address the resistance question, providing possible explanations on how behavior can be robust to such phenomena [18, 19, 20, 21].

Here, we focus on the mechanistic question, and leverage analyses of drift statistics for this purpose. Specifically, recent studies showed that representational drift in the CA1 is driven by active experience [22]. Namely, rate maps decorrelate more when mice are active for a longer time in a given context. This implies that drift is not just a passive process, but rather an active learning one. As drift seems to occur after an adequate representation has formed, it seems fitting to model it as a form of a continuous learning process.

This approach has been recently explored by [23, 24]. They considered continuous learning in noisy, overparameterized neural networks. Because the system is overparameterized, a manifold of zero-loss solutions exists. [23] showed that for feedforward neural networks (FNNs) trained using Hebbian learning with added parameter noise, neurons change their tuning over time. This was due to an *undirected* random walk within the manifold of solutions. The coordinated drift of neighboring place fields was used as evidence to support this view. The phenomenon of undirected motion within the space of solutions seems plausible, as all members of this space achieve equally good performance (Fig 1A left). However, there may be other properties of the solutions (Fig 1B) that vary along this manifold, which could potentially bias drift in a certain direction (Fig 1A right). It is likely that the drift observed in

experiments is a combination of both an undirected and directed movement. We will now introduce theoretical results from machine learning that support the possibility of directed drift.

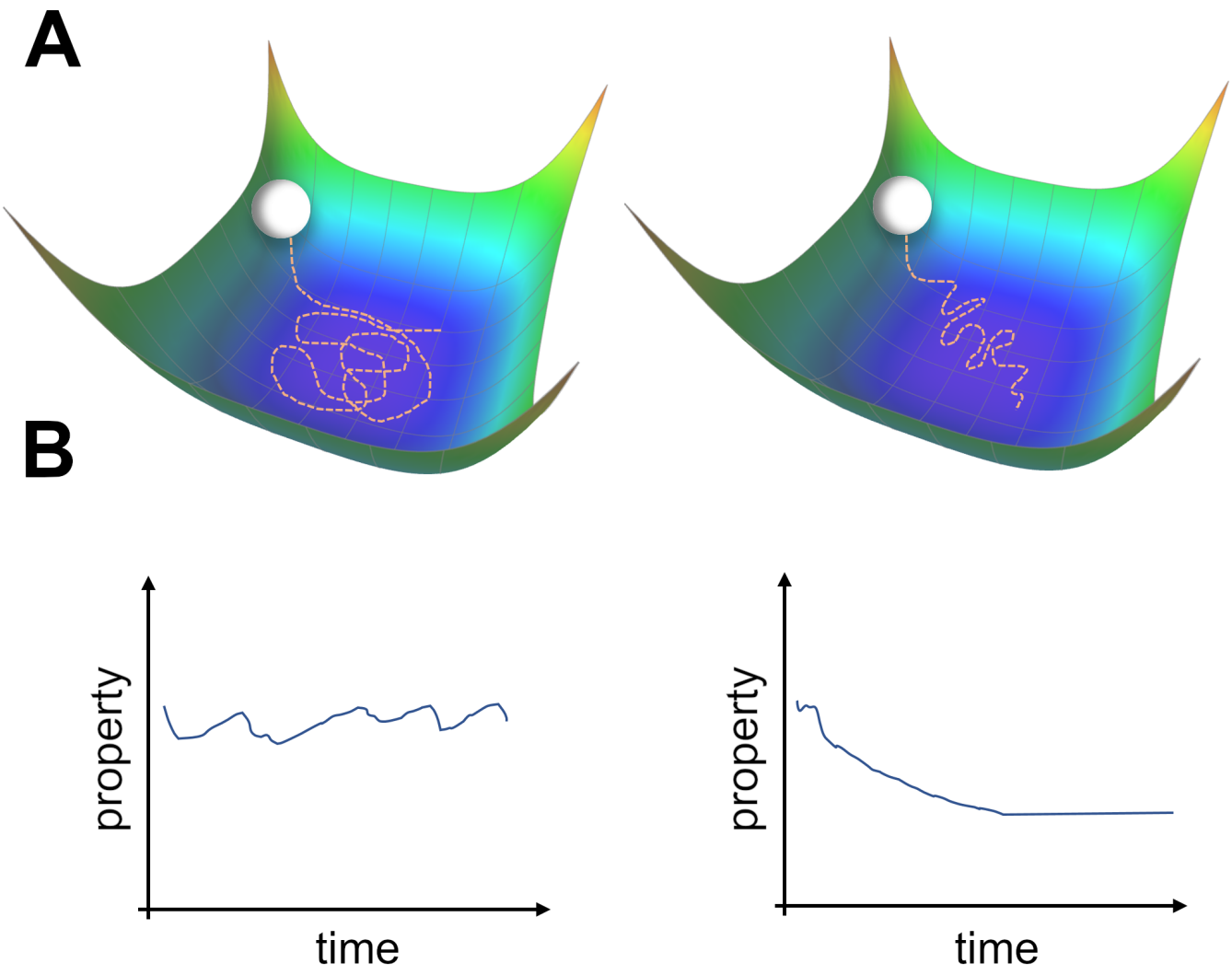

Figure 1: Two types of possible movements within the solution space. (A) Two options of how drift may look in the solution space. Random walk within the space of equally good solutions that is either undirected (left) or directed (right). (B) The qualitative consequence of the two movement types. For an undirected random walk, all properties of the solution will remain roughly constant (left). For the directed movement there should be a given property that is gradually increasing or decreasing (right).

Recent work provided a tractable analytical framework for the learning dynamics of Stochastic Gradient Descent (SGD) with added noise and an overparameterized regime [25, 26]. These studies

showed that, after the network has converged to the zero-loss manifold, a second-order effect biases the random walk along a specific direction within this manifold. This direction reduces an implicit regularizer, determined by the type of noise the network is exposed to. The regularizer is related to the Hessian of the loss – a measure of the flatness of the loss landscape in the vicinity of the solutions. Since this directed movement is a second-order effect, its timescale is orders of magnitude larger than that of the initial convergence.

Consider a biological neural network performing a task. The ML implicit regularization mentioned above requires three components: an overparameterized regime, noise, and SGD. Both biological and artificial networks possess a large number of synapses, or parameters, and hence can reasonably be expected to be overparameterized. Noise can emerge from the external environment or from internal biological elements. It is not reasonable to assume that a precise form of gradient descent is implemented in the brain [27], thereby casting doubt on the third element. Nevertheless, biologically plausible rules could be considered as noisy versions of gradient descent, as long as there is a coherent improvement in performance [28, 29]. Motivated by this analogy, we explore representational drift in models and experimental data.

Because drift is commonly observed in spatially-selective cells, we base our analysis on a model which has been shown to contain such cells [30]. Specifically, we trained artificial neural networks on a predictive coding task in the presence of noise. In this task, an agent moves along a linear track while receiving visual input from the walls, such that the goal is to predict the subsequent input. We observed that neurons became tuned to the latent variable, which is position, in accordance with previous results [30]. We continued training and found that in addition to the gradual change of tuning curves, similar to [23], we witnessed that the number of active neurons decreased slowly while their tuning specificity increased. These results align with recent experimental observations [22]. Finally, we demonstrated the connection between this sparsificiation effect and changes to the Hessian, in accordance with ML theory.

Results

Spontaneous sparsification in a predictive coding network

To model representational drift in the CA1 area, we chose a simple model that could give rise to spatially-tuned cells [30]. In this model, an agent traverses a corridor while slightly modulating its angle with respect to the main axis (Fig 2A). The walls are textured by a fixed smooth noisy signal, and the agent receives this as input according to its current field of view. The model itself is a single hidden layer feedforward network, with the velocity and visual field as inputs. The desired output is the predicted visual input in the next time step. The model equations are given by:

$$\hat{\mathbf{y}}_t = \sigma(\mathbf{x}_t\mathbf{m}^T + \mathbf{b})\mathbf{n}^T, \tag{1}$$

where $\mathbf{m}$ and $\mathbf{n}$ are the input and output matrices respectively, $\mathbf{b}$ is the bias vector, and σ is the ReLU activation function. The task is for the network's output, $\mathbf{y}$, to match the visual input, $\mathbf{x}$ of the following time step, resulting in the following loss function:

$$f(\mathbf{m}, \mathbf{n}, \mathbf{b}) \;=\; \mathbb{E}_t(\hat{\mathbf{y}}_t - \mathbf{x}_{t+1})^2. \tag{2}$$

We train the network using Gradient Descent (GD), while adding update noise to the learning dynamics:

$$\theta_{\tau+1} = \theta_\tau - \eta\frac{\partial f(\theta_\tau)}{\partial \theta_\tau} + \xi_\tau^{update}, \tag{3}$$

where $\theta = (\mathbf{m}, \mathbf{n}, \mathbf{b})$ is the vectorized parameters-vector, τ is the current training step and ξ_τ^{update} is Gaussian noise. We let the network converge to a good solution, demonstrated by a loss plateau, and continue training for an additional period. Note that this additional period can be orders of magnitude longer than the initial training period. The network quickly converged to a low loss and stayed at the same loss during the additional training period (Fig 2B). Surprisingly, when looking at the activity within the hidden layer, we noticed that it slowly became sparse. This sparsification did not hurt performance,

because individual units became more informative, as quantified by the average mutual information between unit activity and the position of the agent (Fig 2C). When looking at the rate maps of neurons, i.e. their tuning to position, one can observe an image similar to representational drift observed in experiments [5] – namely that neurons changed their tuning over time (Fig 2D). Additionally, their tuning specificity increased in accordance with the information increase. By observing the correlation matrix of the rate maps over time, it is apparent that there was a gradual change that slowed down (Fig 2E). To summarize, we observed a spontaneous sparsification over a timescale much longer than the initial convergence, without introducing any explicit regularization. This is comparable to experimental data from [22], where indeed drift was characterized by a decrease in the fraction of active place cells, and an increase in cells' information while the decoding error for the position of the mouse stayed relatively constant (Fig 2F). Another recent study further demonstrated an increase in information over days [31].

Generality of the phenomenon

To explore the sensitivity of our results to specific modeling choices, we systematically varied many of them (Fig 3A). Specifically, we replaced the task with either a simplified predictive coding, random mappings or smoothed random mappings. Noise was introduced to the outputs (label noise), instead of the update noise. We simulated different activation functions. Perhaps most important, we varied the learning rules, as SGD is not a biologically plausible one. We used both Adam [32] and RMSprop [33], from the ML literature. We also used Stochastic Error-Descent (SED) [34], which does not require gradient calculation and is more biologically plausible (5). All cases demonstrated an initial, fast, phase of convergence to low loss, followed by a much slower phase of directed random motion within the low-loss space.

The results of the simulations supported our main conclusion, though several qualitative phenomenons could be observed. First of all, sparsification dynamics were not sensitive to most of the parameters. The main qualitative difference observed was that the timescales could vary by orders of magnitude as a function of the noise scale (Fig 3B bottom). Note that we calculate the timescale of sparsification by fitting an exponential curve to the fraction of active units over time, and take the time constant of the fitted exponential. Additionally, apart from simulations that did not converge due to too big timescales, the final sparsity was the same for all networks of the same size (Fig 3B top), in

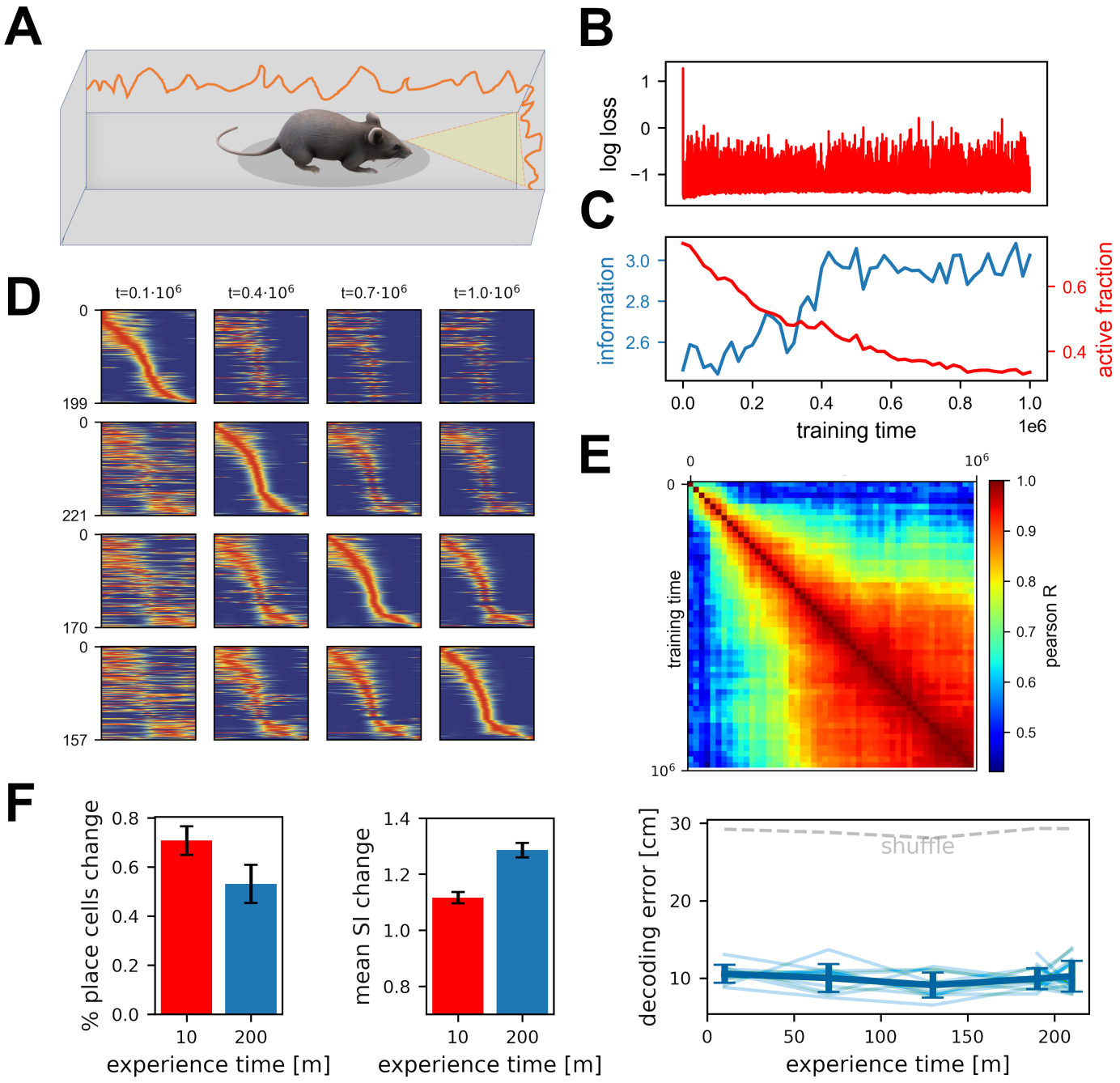

Figure 2: Noisy learning leads to spontaneous sparsification. (A) Illustration of an agent in a corridor receiving high-dimensional visual input from the walls. (B) Log loss as a function of training steps, log loss of 0 corresponds to a mean estimator. The loss rapidly decreases, and then remains roughly constant. (C) Information (blue) and fraction of units with non-zero activation for at least one input (red) as a function of training steps. (D) Rate maps sampled at four different time points. Maps in each row are sorted according to a different time point. Sorting is done based on the peak tuning value to the latent variable. (E) Correlation of rate maps between different time points along training. Only active units are used. (F) Figures reproduced from [22] where mice spent different amount of time in two environments. Fraction of place cells in the beginning relative to the end of the experiment (left), average Spatial Information (SI) per cell in the beginning relative to the end of the experiment (middle) and the decoding error for the position of the mouse (right).

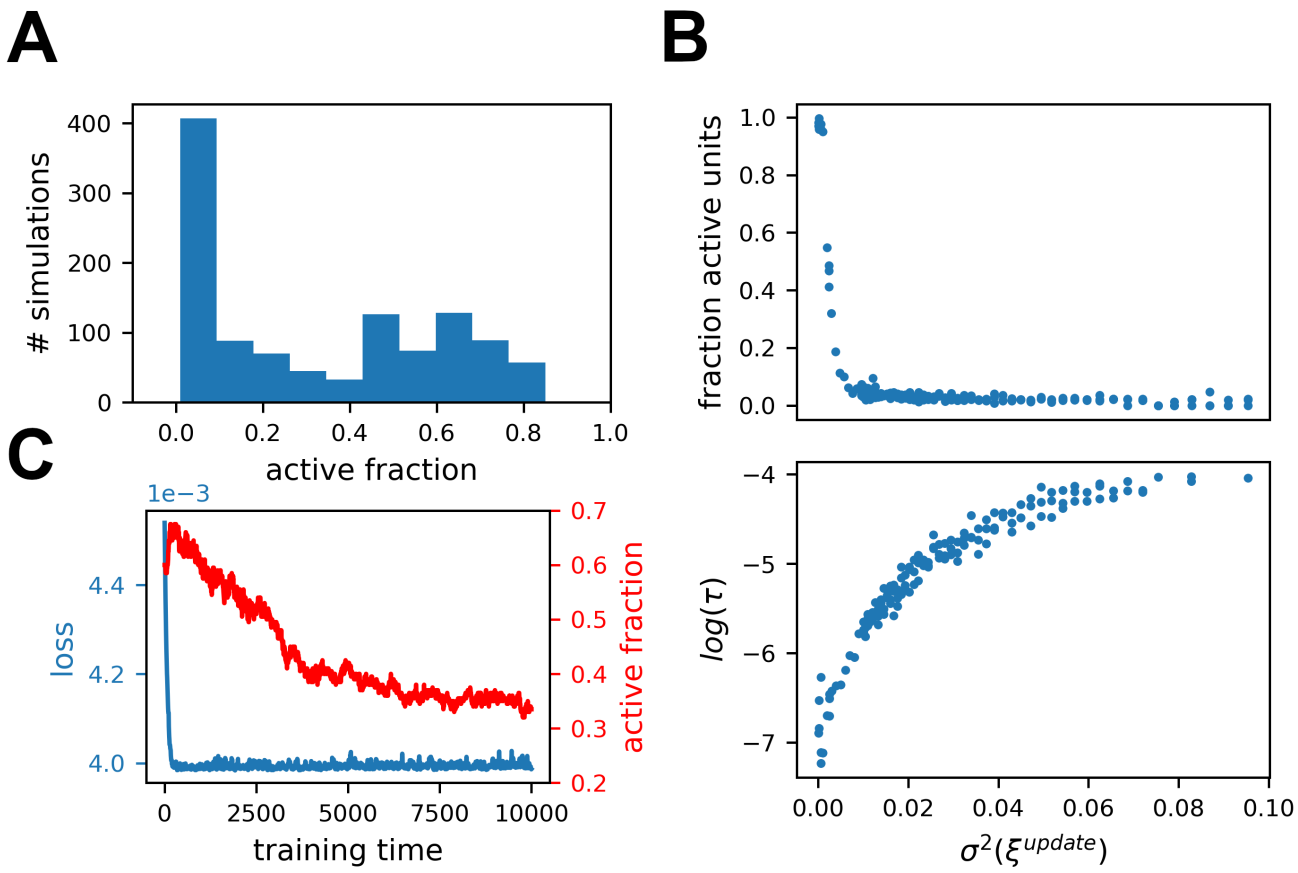

Figure 3: Generality of the results. Summary of 1117 simulations with various parameters (see Table 1). (A) Histogram of fraction of active units after 10^7 training steps for each simulation. (B) Subset of 178 simulations with the same parameters and varying noise variance, each point represents a single simulation. Fraction of active units as a function of the variance of the noise (top), the log of sparsification time scale as a function of the variance of the noise (bottom). (C) Learning a similarity matching task with Hebbian and anti-Hebbian learning with published code from [23]. Performance of the network (blue) and fraction of active units (red) as a function of training steps. Note that the loss axis does not start at zero, and the dynamic range is small.

accordance with results from [23]. In a sense, once noise is introduced the network is driven to maximal sparsification. For Adam, RMSprop and SED sparsification ensued in the absence of any added noise. For SED the explanation is straightforward, as the parameter updates are driven by noise. For Adam and RMSprop, we suggest that in the vicinity of the zero-loss manifold, the second moment acts as noise. For label noise, the dynamics were qualitatively different, the fraction of active units did not reduce, but the activity of the units did sparsify. In some cases, the networks quickly collapsed to a sparse

solution, most likely as a result of the learning rate being too high, in relation to the input statistics [35]. Importantly, for GD without noise, there was no change after the initial convergence.

As a further test of the generality of this phenomenon, we consider the recent simulation from [23]. The learning rule used in this work was very different from the ones we applied. We, therefore, simulated that network using the published code. We found the same type of dynamics as shown above, namely that the network initially converged to a good solution followed by a longer period of sparsification (Fig 3C). Note that in The original publication [23] the focus was on the stage following this sparsification, in which the network indeed maintained a constant fraction of active cells.

In conclusion, we see that noisy learning leads to three phases under rather general conditions. First, fast learning of the task and convergence to the manifold of low-loss solutions. The second phase is directed movement on this manifold driven by a second-order effect of implicit regularization. The third phase is an undirected random walk within the sub-manifold of low loss and maximum regularization.

Mechanism of sparsification

What are the mechanisms that give rise to sparsification? As illustrated in Fig. 1, different solutions in the zero-loss manifold might vary in some of their properties. The specific property suggested from theory [25] is the flatness of the loss landscape in the vicinity of the solution. This can be demonstrated with a simple example. Consider a two-dimensional loss function. The function is shaped like a valley with a continuous one-dimensional zero-loss manifold at it's bottom (Fig 4A). Crucially, the loss on the entire manifold is exactly zero, while the vicinity of the manifold becomes systematically flatter in one direction. We simulated gradient descent with added noise on this function from a random starting point (red dot). The trajectory quickly converged to the zero-loss manifold, and began a random walk on it. This walk was clearly biased towards the flatter area of the manifold, as can be seen by the spread of the trajectory. This bias could be comprehended by noting that the gradient was orthogonal to the contour lines of the loss, and therefore had a component directed towards the flat region.

In higher dimensions, flatness is captured by the eigenvalues of the Hessian of the loss. Because these eigenvalues are a collection of numbers, different scenarios could lead to minimizing different aspects of this collection. Specifically, according to [25], update noise should regularize the sum of the log of the non-zero eigenvalues while label noise should do the same for the sum of eigenvalues. In our

predictive coding example, where update noise was added, each inactivated unit translates into a set of zero-rows in the Hessian, and thus also into a set of zero-eigenvalues (Fig 4B). The slope of the regularizer approaches infinity as the eigenvalue approaches zero, and thus small eigenvalues are driven to zero much faster than large eigenvalues (Fig 4C). So in this case, update noise leads to an increase in the number of zero eigenvalues, which are manifested as a sparse solution. Another, perhaps more intuitive, way to understand these results is that units below the activation threshold are insensitive to noise perturbations. In other scenarios, in which we simulated with label noise, we indeed observed a gradual decrease in the sum of eigenvalues (Fig 4D).

Discussion

We showed that representational drift could arise from ongoing learning in the presence of noise, after a network has already reached good performance. We suggest that learning is divided into three overlapping phases: a fast initial phase, where good performance is achieved, a second slower phase in which *directed* drift along the low-loss manifold leads to an implicit regularization and finally, a third *undirected* phase ensues once the regularizer is minimized. In our results, the directed component was associated with sparsification of the neural code, a phenomenon we also observed in experimental data.

Interpreting drift as a learning process has recently been suggested by [23, 24]. Both studies focused on the final phase in which the statistics of the representations were constant. Experimentally, [7] reported a decrease in activity at the beginning of the experiment, which they suggested was correlated with some behavioral change, but we believe it could also be a result of the directed drift phase. [36] also reported a slow directed change in representation long after familiarity with the stimuli. There is another consequence of the timescale separation. Unlike in the setting of drift experiments, natural environments are never truly constant. Thus, it is possible that the second phase of learning never stops because the task is slowly changing. This would imply that the second, directed, phase may be the natural regime in which neural networks reside.

Here, we reported directed drift in the space of solutions of neural networks. This drift could be observed by examining changes to the representation of external world variables, and hence is related to the phenomenon of representational drift. Note, however, that representations are not a full description

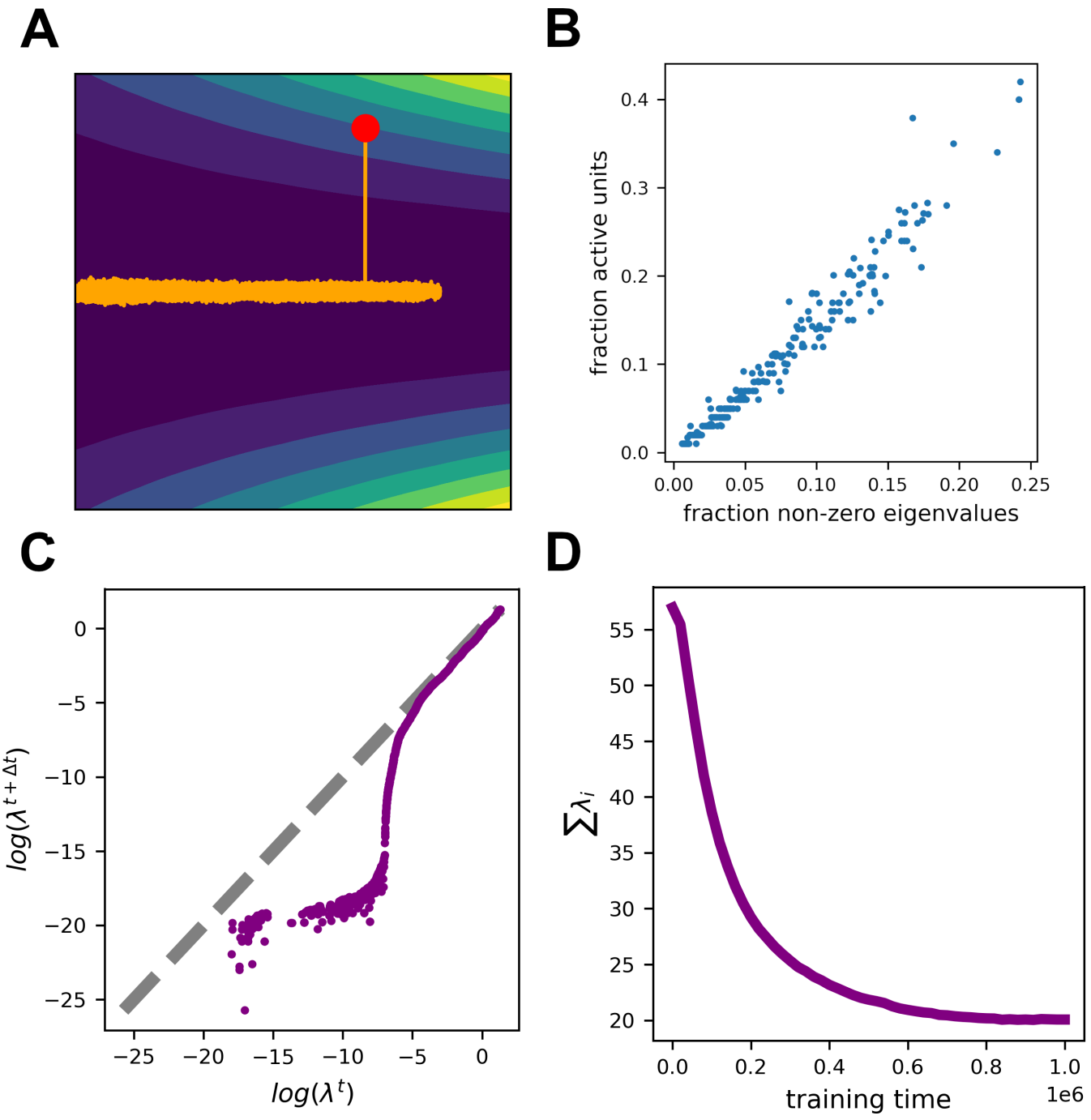

Figure 4: Noisy learning leads to a flat landscape. (A) Gradient Descent dynamics over a two-dimensional loss function with a one-dimensional zero-loss manifold. Note that the loss is identically zero along the horizontal axis, but the left area is flatter. The orange trajectory begins at the red dot. Note the asymmetric extension into the left area. (B) Fraction of active units as a function of the number of non-zero eigenvalues. (C) Log of non-zero eigenvalues at two consecutive time points. Note that eigenvalues do not correspond to one another when calculated at two different time points, and this plot demonstrates the change in their distribution rather than changes in eigenvalues corresponding to specific directions. The distribution of larger eigenvalues hardly changes, while the distribution of smaller eigenvalues is pushed to smaller values. (D) Sum of the Hessian's eigenvalues as a function of time for learning with label noise.

of a network's behavior [37]. The statistics of representational changes can be used as a window into changes of network dynamics and function.

The phenomenon of directed drift is very robust to various modeling choices, and also consistent with recent theoretical results [25, 26] The details of the direction of the drift, however, are dependent on specific choices. Specifically, which aspects of the Hessian are minimized during the second phase of learning, as well as the timescale of this phase, depend on the specifics of the learning rule and the noise in the system. This suggests an exciting opportunity – inferring the learning rule of a network from the statistics of representational drift.

Our explanation of drift invoked the concept of a low-loss manifold – a family of network configurations that have identical performance on a task. The definition of low-loss, however, depends on the specific task and context analyzed. Challenging a system with new inputs could dissociate two configurations that otherwise appear identical [38]. It will be interesting to explore whether various environmental perturbations could uncover the motion along the low-loss manifold in the CA1 population. For instance, remapping was interpreted as an indicator of the detection of a context switch [39]. One can therefore speculate that the probability for remapping given the same environmental change will systematically vary as the network moves to flatter areas of the loss landscape.

Machine learning has been suggested as a model tool for neuroscience research [40, 41, 42]. However, the implicit regularization in ML has not been studied to explain representational drift in neuroscience, and may have been done without awareness of this phenomenon. It's worth noting that this isn't a phenomenon specific to neural networks, but rather a general property of overparameterized systems that optimize a cost function. Importing insights from this domain into neuroscience shows the utility of studying general phenomena in systems that learn. For example, another complex learning system in which a similar idea has been proposed is evolution – "survival of the flattest" suggests that, under a high mutation rate, the fittest replicators are not just the ones with the highest fitness, but also with a flat fitness function which is more robust to mutations [43]. One can hope that more such insights will arise as we open our eyes.

Materials and methods

Predictive coding task

The agent is moving in an arena of size (L_x, L_y), with constant velocity in the y direction of V_0. The agent's heading direction is θ and it changes at every time step by $\Delta\theta \sim G(0, \sigma_\theta^2)$, the agent's visual field has an angle θ_{vis} and is represented as a vector of size L_{vis}. The texture of the walls is generated from a random Gaussian vector of size $L_{walls} = 2(L_x + L_y)L_{vis}$, smoothed with a Gaussian filter with $\sigma^2 = K_{smooth}L_{walls}$. At each time step the agent receives the visual input from the walls, determined by the intersection points of it's visual field with the walls. When the agent reaches a distance of $L_y L_{buffer}$ from the wall, it turns to the opposite direction.

Tuning properties of units

For each unit we calculated a tuning curve. We divided the arena into 100 equal bins and computed the number of time steps in each bin and the mean unit activation. We then obtained the tuning curve by dividing the mean activity for each bin by the occupancy. We treated movement in each direction as a separate location. We calculated the spatial information (SI) of the tuning curves for each unit:

$$SI = \sum_i p_i \frac{r_i}{\bar{r}} \log_2 \frac{r_i}{\bar{r}} \tag{4}$$

where i is the index of the bin, p_i is the probability of being in the bin, r_i is the value of the tuning curve in the bin and $\bar{r}$ is the unit's mean activity rate. Active unit was defined as a unit with non-zero activation for at least one input.

Simulations

For the random simulations, we train each network for 10^7 training steps while choosing random learning algorithm and parameters. The ranges and relevant values of parameters are specified in Table 1. For Adam and SED there was no added noise.

Table 1: Parameter ranges for random simulations.

Parameter	Possible values
learning algorithm	{SGD, Adam, SED}
noise type	{update, label}
number of samples	[1,20]
initialization regime	{lazy, rich}
task	{abstract predictive, random, random smoothed}
input dimension	[1,20]
output dimension	[1,20]
noise variance (label/update)	[0.1,1]/[0.01,0.1]
hidden layer size	100

Stochastic Error Descent

The equation for parameter updates under this learning rule is given by:

$$\theta_{\tau+1} = \theta_\tau - \eta(f(\theta_\tau + \xi_\tau) - f(\theta_\tau))\xi_\tau \tag{5}$$

In this learning rule, the parameters are randomly perturbed at each training step by a Gaussian noise denoted by ξ_τ and then updated in proportion to the change in loss.

Label noise

Label noise is introduced to the loss function given by the following formula:

$$f(\mathbf{x}_t) \;=\; (\hat{\mathbf{y}}_t - \mathbf{x}_{t+1} + \xi_\tau^{label})^2, \tag{6}$$

where ξ_τ^{label} is Gaussian noise.

Gradient descent dynamics around the zero-loss manifold

The function we used for the two-dimensional example was given by:

$$L(x,y) \;=\; (xy)^2, \tag{7}$$

which has zero loss on the x and y axes. For small enough update noise, GD will converge to the vicinity of this manifold (the axes). We consider a point on the x axis: $(x_0, 0)$, and calculate the direction of the gradient near that point. Because we are interested in motion along the zero-loss manifold, we consider a small perturbation in the orthogonal direction $(x_0, 0 + \Delta y)$ where $x_0 >> 1$ and $|\Delta y| << 1$. Any component of the gradient in the x direction will lead to motion along the manifold. The update step at this point is given by:

$$-\nabla L(x_0, 0 + \Delta y) \;=\; -2x_0 \begin{pmatrix} (\Delta y)^2 \\ x_0 \Delta y \end{pmatrix}. \tag{8}$$

One can observe that the step has a large component in the y direction, quickly returning to the manifold. There is also a smaller component in the x direction, reducing the value of x. Reducing x also reduces the Hessian's eigenvalues:

$$H_L(x_0, 0) = 2 \begin{pmatrix} 0 & 0 \\ 0 & x_0^2 \end{pmatrix} \tag{9}$$

$$\lambda_{1,2} = \{0, x_0^2\}, v_{1,2} = \{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \}. \tag{10}$$

Thus, it becomes clear that the trajectory will have a bias that reduces the curvature in the y direction.

For general loss functions and various noise models, rigorous proofs can be found in [25], and a different approach can be found in [26]. Here, we will briefly outline the intuition for the general case. Consider again the update rule for GD:

$$\theta \leftarrow \theta - \eta \nabla L(\theta). \tag{11}$$

In order to understand the dynamics close to the zero-loss manifold, we consider a point θ, for which $L(\theta) = 0$ expand the loss around it:

$$L(\theta + \delta\theta) = L(\theta) + \nabla^T L(\theta)\delta\theta + \frac{1}{2}\delta\theta^T H \delta\theta. \tag{12}$$

We can then take the gradient of this expansion with respect to θ:

$$\begin{aligned} \nabla_\theta L(\theta + \delta\theta) &= \nabla_\theta L(\theta) + \nabla_\theta \nabla_\theta^T L(\theta)\delta\theta + \nabla_\theta(\frac{1}{2}\delta\theta^T H \delta\theta) & (13) \\ &= 0 + H\delta\theta + \nabla_\theta(\frac{1}{2}\delta\theta^T H \delta\theta). & (14) \end{aligned}$$

The first term is zero, because the gradient is zero on the manifold. The second term is the largest one, as it linear in $\delta\theta$. Note that the Hessian matrix has zero eigenvalues in directions on the zero-loss manifold, and non-zero eigenvalues in other directions. Thus, the second term corresponds to projecting $\delta\theta$ in a direction that is orthogonal to the zero-loss manifold. The third term can be interpreted as the gradient of some auxiliary loss function. Thus, we expect gradient descent to minimize this new loss, which corresponds to a quadratic form with the Hessian. This is the reason for the implicit regularization along the manifold. Note that the auxiliary loss function is defined by $\delta\theta$, and thus different noise statistics will correspond, on average, to different implicit regularizations. In conclusion, the update step will have a large component that moves the parameter vector towards the zero-loss manifold, and a small component that moves the parameter vector on the manifold in a direction that minimizes some measure of the Hessian.

Hessian and sparseness

In the main text, we show that the implicit regularization of the Hessian leads to sparse representations. Here, we show this relationship for a single-hidden layer feed-forward neural network with ReLU activation and Mean Squared Error loss:

$$f(\mathbf{x}_i) = \sigma(\mathbf{x}_t \mathbf{m}^T + b)\mathbf{n}^T \tag{15}$$

The gradient and Hessian at the zero-loss manifold are given by [44]:

$$\nabla_\theta f(\mathbf{x}_i) = \begin{pmatrix} \frac{\partial f}{\partial \mathbf{m}} \\ \frac{\partial f}{\partial \mathbf{b}} \\ \frac{\partial f}{\partial \mathbf{n}} \end{pmatrix} = \begin{pmatrix} \mathbf{n} \odot \mathbb{1}(\mathbf{x}_i;\theta) \otimes \mathbf{x}_i \\ \mathbf{n} \odot \mathbb{1}(\mathbf{x}_i;\theta) \\ (\mathbf{x}_i \cdot \mathbf{n}^T + \mathbf{b}) \odot \mathbb{1}(\mathbf{x}_i;\theta) \end{pmatrix} \tag{16}$$

$$\nabla^2_\theta L(\mathbf{x};\theta) = \sum_i \nabla_\theta f(\mathbf{x}_i) \nabla_\theta f(\mathbf{x}_i)^T, \tag{17}$$

where $\mathbb{1}(\mathbf{x}_i;\theta)$ is an indicator vector denoting whether each unit is active for some input $\mathbf{x}_i$. Sparseness means that a unit has become inactive for all inputs. All the partial derivatives of input, output and bias weights associated with such a unit are zero, and thus the relevant rows of the Hessian are zero as well. Thus, every inactive unit leads to several zero eigenvalues.

Acknowledgments

We thank Ron Teichner and Kabir Dabholkar for comments on the manuscript. This research was supported by the ISRAEL SCIENCE FOUNDATION (grants Nos. 2655/18 and 2183/21 to DD, and 1442/21to OB), by the German-Israeli Foundation (GIF I-1477-421.13/2018) to DD, by a grant from the US-Israel Binational Science Foundation (NIMH-BSF CRCNS BSF:2019807, NIMH:R01 MH125544-01 to DD), by an HFSP research grant (RGP0017/2021) to OB, A Rappaport Institute Collaborative research grant to DD, by Israel PBC-VATAT and by the Technion Center for Machine Learning and Intelligent Systems (MLIS) to DD and OB, by the Prince Center for the Aging Brain, and by a University of Michigan – Israel Partnership for Research and Education Collaborative Research stipend to DK. (data science)

References

[1] John O'keefe and Lynn Nadel. The hippocampus as a cognitive map. *Behavioral and Brain Sciences*, 2(4):487–494, 1979.

[2] John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 1971.

[3] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[4] Bruce L McNaughton, SJY Mizumori, CA Barnes, BJ Leonard, M Marquis, and EJ Green. Cortical representation of motion during unrestrained spatial navigation in the rat. *Cerebral Cortex*, 4(1):27–39, 1994.

[5] Yaniv Ziv, Laurie D Burns, Eric D Cocker, Elizabeth O Hamel, Kunal K Ghosh, Lacey J Kitch, Abbas El Gamal, and Mark J Schnitzer. Long-term dynamics of ca1 hippocampal place codes. *Nature neuroscience*, 16(3):264–266, 2013.

[6] Laura N. Driscoll, Noah L. Pettit, Matthias Minderer, Selmaan N. Chettih, and Christopher D. Harvey. Dynamic Reorganization of Neuronal Activity Patterns in Parietal Cortex. *Cell*, 170(5):986–999, 8 2017.

[7] Daniel Deitch, Alon Rubin, and Yaniv Ziv. Representational drift in the mouse visual cortex. *bioRxiv*, page 2020.10.05.327049, 10 2020.

[8] Carl E Schoonover, Sarah N Ohashi, Richard Axel, and Andrew J P Fink. Representational drift in primary olfactory cortex. *Nature*, 594, 2021.

[9] Ziv Y. Geva N., Rubin A. Differential effects of time and experience on hippocampal representational drift. *[Poster]. In: Cosyne Convention, 2022, March 17-20, Lisbon, Portugal*, 2022.

[10] William A Liberti, 3rd, Tobias A Schmid, Angelo Forli, Madeleine Snyder, and Michael M Yartsev. Publisher correction: A stable hippocampal code in freely flying bats. *Nature*, 606(7914):E6, June 2022.

[11] Sadra Sadeh and Claudia Clopath. Contribution of behavioural variability to representational drift. *Elife*, 11:e77907, 2022.

[12] Michael E Rule, Timothy O'Leary, and Christopher D Harvey. Causes and consequences of representational drift. *Curr. Opin. Neurobiol.*, 58:141–147, October 2019.

[13] Laura N Driscoll, Lea Duncker, and Christopher D Harvey. Representational drift: Emerging theories for continual learning and experimental future directions. *Current Opinion in Neurobiology*, 76:102609, 2022.

[14] Noam E Ziv and Naama Brenner. Synaptic tenacity or lack thereof: spontaneous remodeling of synapses. *Trends in neurosciences*, 41(2):89–99, 2018.

[15] Alon Rubin, Nitzan Geva, Liron Sheintuch, and Yaniv Ziv. Hippocampal ensemble dynamics timestamp events in long-term memory. *elife*, 4:e12247, 2015.

[16] Kyle Aitken, Marina Garrett, Shawn Olsen, and Stefan Mihalas. The geometry of representational drift in natural and artificial neural networks. *PLOS Computational Biology*, 18(11):e1010716, 2022.

[17] David Kappel, Stefan Habenschuss, Robert Legenstein, and Wolfgang Maass. Network plasticity as bayesian inference. *PLoS computational biology*, 11(11):e1004485, 2015.

[18] Uri Rokni, Andrew G Richardson, Emilio Bizzi, and H Sebastian Seung. Motor learning with unstable neural representations. *Neuron*, 54(4):653–666, 2007.

[19] Lee Susman, Naama Brenner, and Omri Barak. Stable memory with unstable synapses. *Nature communications*, 10(1):4441, 2019.

[20] Gianluigi Mongillo, Simon Rumpel, and Yonatan Loewenstein. Intrinsic volatility of synaptic connections—a challenge to the synaptic trace theory of memory. *Current opinion in neurobiology*, 46:7–13, 2017.

[21] Yaroslav Felipe Kalle Kossio, Sven Goedeke, Christian Klos, and Raoul-Martin Memmesheimer. Drifting assemblies for persistent memory: Neuron transitions and unsupervised compensation. *Proceedings of the National Academy of Sciences*, 118(46):e2023832118, 2021.

[22] Dorgham Khatib, Aviv Ratzon, Mariell Sellevoll, Omri Barak, Genela Morris, and Dori Derdikman. Experience, not time, determines representational drift in the hippocampus. *bioRxiv*, pages 2022–08, 2022.

[23] Shanshan Qin, Shiva Farashahi, David Lipshutz, Anirvan M Sengupta, Dmitri B Chklovskii, and Cengiz Pehlevan. Coordinated drift of receptive fields in hebbian/anti-hebbian network models during noisy representation learning. *Nature Neuroscience*, pages 1–11, 2023.

[24] Farhad Pashakhanloo and Alexei Koulakov. Stochastic gradient descent-induced drift of representation in a two-layer neural network. *arXiv preprint arXiv:2302.02563*, 2023.

[25] Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on learning theory*, pages 483–513. PMLR, 2020.

[26] Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after sgd reaches zero loss?–a mathematical framework. *arXiv preprint arXiv:2110.06914*, 2021.

[27] Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, and Zhouhan Lin. Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*, 2015.

[28] Yuhan Helena Liu, Arna Ghosh, Blake Richards, Eric Shea-Brown, and Guillaume Lajoie. Beyond accuracy: generalization properties of bio-plausible temporal credit assignment rules. *Advances in Neural Information Processing Systems*, 35:23077–23097, 2022.

[29] Owen Marschall, Kyunghyun Cho, and Cristina Savin. A unified framework of online learning algorithms for training recurrent neural networks. *The Journal of Machine Learning Research*, 21(1):5320–5353, 2020.

[30] Stefano Recanatesi, Matthew Farrell, Guillaume Lajoie, Sophie Deneve, Mattia Rigotti, and Eric Shea-Brown. Predictive learning as a network mechanism for extracting low-dimensional latent space representations. *Nature Communications*, 12(1), 2021.

[31] Liron Sheintuch, Alon Rubin, and Yaniv Ziv. Bias-free estimation of information content in temporally sparse neuronal activity. *PLoS computational biology*, 18(2):e1009832, 2022.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[34] Gert Cauwenberghs. A fast stochastic error-descent algorithm for supervised learning and optimization. *Advances in neural information processing systems*, 5, 1992.

[35] Rotem Mulayoff, Tomer Michaeli, and Daniel Soudry. The implicit bias of minima stability: A view from function space. *Advances in Neural Information Processing Systems*, 34:17749–17761, 2021.

[36] Nghia D Nguyen, Andrew Lutas, Jesseba Fernando, Josselyn Vergara, Justin McMahon, Jordane Dimidschstein, and Mark L Andermann. Cortical reactivations predict future sensory responses. *bioRxiv*, pages 2022–11, 2022.

[37] Romain Brette. Is coding a relevant metaphor for the brain? *Behavioral and Brain Sciences*, 42:e215, 2019.

[38] Elia Turner, Kabir V Dabholkar, and Omri Barak. Charting and navigating the space of solutions for recurrent neural networks. *Advances in Neural Information Processing Systems*, 34:25320–25333, 2021.

[39] Honi Sanders, Matthew A Wilson, and Samuel J Gershman. Hippocampal remapping as hidden state inference. *Elife*, 9:e51140, 2020.

[40] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770, 2019.

[41] Adam H Marblestone, Greg Wayne, and Konrad P Kording. Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, page 94, 2016.

[42] Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1):55–67, 2021.

[43] Francisco M Codoñer, José-Antonio Darós, Ricard V Solé, and Santiago F Elena. The fittest versus the flattest: experimental confirmation of the quasispecies effect with subviral pathogens. *PLoS pathogens*, 2(12):e136, 2006.

[44] Mor Shpigel Nacson, Rotem Mulayoff, Greg Ongie, Tomer Michaeli, and Daniel Soudry. The implicit bias of minima stability in multivariate shallow reLU networks. *inproceedings*, 2023.