# Langevin Learning Dynamics in Lazy and Non-Lazy Wide Neural Networks

**Yehonatan Avidan**[1,2], **Haim Sompolinsky**[1,2,3]
[1]Racah Institute of Physics, The Hebrew University,
[2]Edmond and Lily Safra Center for Brain Sciences, The Hebrew University,
[3]Center for Brain Science, Harvard University

## Abstract

Langevin dynamics—gradient descent with additive stochastic noise—provides a powerful framework for learning dynamics in deep neural networks, bridging deterministic optimization and statistical inference in deep neural networks. It has been shown to unify two prominent theories for wide networks: the Neural Tangent Kernel (NTK), which assumes linearized gradient descent dynamics, and the Bayesian Neural Network Gaussian Process (NNGP), which treats learning as posterior inference. In this work, we extend the framework to compare lazy and non-lazy learning in linear networks, analyzing how different parameters affect the learning dynamics of both the predictor and the kernel in each regime. We show that in the non-lazy case, the network is more resilient to noise and to small initial condition.

## 1. Introduction

The success of deep learning has spurred intense interest in understanding the underlying dynamics of neural network training. A central challenge is to connect optimization algorithms with principles that govern statistical inference [17]. Langevin dynamics—gradient descent with additive noise [8, 33]—provides a natural bridge between these perspectives, interpolating between initial learning dominated by deterministic gradient and convergence to equilibrium equivalent to Bayesian inference. In this work, we use Langevin dynamics to study neural networks in the infinite width regime. We previously used this framework to unify two prominent theoretical approaches to the kernel regime [1]: the Neural Tangent Kernel (NTK) [15], which assumes linearized gradient descent, and the Neural Network Gaussian Process (NNGP) [7, 16, 22, 34], which describes the long-time behavior dominated by sampling from a posterior distribution. We extend the theory to compare learning in lazy (kernel regime) and non-lazy (feature learning regime) [35, 36] networks under Langevin dynamics in linear networks. We analyze how noise, initialization, and network scaling interact to shape the input-output function dynamics.

## 2. Notations and Setup for the Dynamical Theory

We consider a fully connected DNN with an input $\mathbf{x} \in \mathbb{R}^{N_0}$, $L$ hidden layers. and a single output $f$ (i.e. the predictor). The input-output function is given by:

$$f(\Theta, \mathbf{x}) = N_L^{-\gamma}\, \mathbf{a} \cdot \mathbf{x}^L(\mathbf{x}), \quad \mathbf{a} \in \mathbb{R}^{N_L} \tag{1}$$

$$\mathbf{x}^l(\mathbf{x}) = \phi\left(N_{l-1}^{-1/2}\, \mathbf{W}^l \cdot \mathbf{x}^{l-1}(\mathbf{x})\right), \quad \mathbf{x}^\ell \in \mathbb{R}^{N_\ell}, \quad \ell = 1 \dots L \tag{2}$$

$N_l$ denotes the number of nodes in hidden layer $l$, and $N_0$ is the input dimension. $\mathbf{a} \in \mathbb{R}^{N_L}$ denotes the linear readout weights and $\mathbf{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$ denotes the hidden layer weights between layers $l-1$ and $l$. $\phi$ is an element-wise nonlinear function of the preactivation vector. The set of all

network parameters are denoted collectively as $\Theta \equiv \left\{\mathbf{W}^1 \ldots \mathbf{W}^L, \mathbf{a}\right\}$. $\mathbf{x}^l$ stands for the activations of the neurons in the $l$-th layer, and $\mathbf{x} \in \mathbb{R}^{N_0}$ represents the input vector to the first layer of the network ($\mathbf{x}^{l=0} \equiv \mathbf{x}$). The training data is a set of $P$ labeled examples $\mathcal{D} : \left\{\mathbf{x}^\mu, y^\mu\right\}_{\mu=1,\cdots,P}$ where $\mathbf{x}^\mu \in \mathbb{R}^{N_0}$ is a training data point, and $y^\mu \in \mathbb{R}$ is the target label of $\mathbf{x}^\mu$. It is convenient to define a vector that contains all the label values $Y \in \mathbb{R}^P$ and a vector of the predictor values on all the training points $f_{\text{train}}(t) \in \mathbb{R}^P$, such that $f^\mu_{\text{train}} = f(\Theta, \mathbf{x}^\mu)$. Importantly, in this work we consider the infinite-width regime [16, 22, 34], namely $N_1, \ldots N_L \to \infty$, but finite $P$.

The normalization factor $N_L^{-\gamma}$ control the level of feature learning in the network, where $\gamma = 1/2$ (standard scaling) has been shown to have no feature learning in the infinite width limit [16].

In contrast $\gamma = 1$ (mean field scaling), has been shown to have strong feature learning, as the readout weights $\mathbf{a}$ and the activations $\mathbf{x}^L(\mathbf{x})$ are forced to align to balance the normalization factor.

We consider the following supervised learning cost function:

$$E\left(\Theta_t|\mathcal{D}\right) = \frac{1}{2}\sum_{\mu=1}^{P}\left(f^\mu_{\text{train}}(t) - y^\mu\right)^2 + \frac{T}{2\sigma^2}\left|\Theta_t\right|^2 \tag{3}$$

We introduce the parameter $T\sigma^{-2}$ as controlling the relative strength of the first term (SE loss) and the regularization term (weight decay) similar to [18, 21]. $\sigma^2$ is equivalent to the variance of the Gaussian prior in a Bayesian framework and $T$ control the level of noise in the dynamics (see below).

We consider gradient descent learning dynamics with an additive noise given by continuous-time Langevin equation. The weights of the system start from an i.i.d. Gaussian initial condition with zero mean and variance $\sigma_0^2$. The weights evolve under gradient descent with respect to the cost function above with a noise term $\xi$:

$$\frac{d}{dt}\Theta_t = -\nabla_\Theta E\left(\Theta_t\right) + \xi\left(t\right) \tag{4}$$

where $\xi\left(t\right)$ has a white noise statistics $\left\langle\xi\left(t\right)\right\rangle = 0, \left\langle\xi\left(t\right)\xi^\top\left(t'\right)\right\rangle = 2IT\delta\left(t-t'\right)$.

Given a distribution of initial weights, the Langevin dynamics defines a time-dependent posterior distribution on weight space, $P_t\left(\Theta\right)$, which converges at long times to an equilibrium Gibbs distribution, $P_{eq}(\Theta) \propto \exp\left(-\frac{1}{T}E(\Theta)\right)$.

In the absence of training signal the Langevin dynamics are a random walk with a quadratic potential (an Orenstein-Ulenbeck process [31]). The induced statistics of $\Theta$ is that of temporally correlated i.i.d Gaussian variables with zero mean

$$\left\langle\Theta_t\right\rangle_0 = 0, \quad \left\langle\Theta_t\Theta_{t'}^\top\right\rangle_0 = m(t,t')I \tag{5}$$

$$m(t,t') = \sigma^2 e^{-T\sigma^{-2}|t-t'|} + \left(\sigma_0^2 - \sigma^2\right)e^{-T\sigma^{-2}(t+t')} \tag{6}$$

where $\left\langle\right\rangle_0$ denotes henceforth averaging over the dynamics induced by the regularization and the noise. As expected, $m(0,0) = \sigma_0^2$. At long times, the second term of Eq.6 representing the transient of the dynamics vanishes and the dominant term is $\sigma^2 e^{-T\sigma^{-2}|t-t'|}$, with no dependence on $\sigma_0^2$.

## 3. Lazy Learning in Nonlinear Deep Network

We present here the results from our previous work [1], which used Markov proximal learning approach (see SI Sec.B) to show that in nonlinear deep network in the lazy learning setup ($\gamma = 1/2$),

the moments of the predictor obey a set of integral equations. The equations describing the second moment for a general nonlinearity are complex, and given in SI Sec.D. Here we bring the equations for the mean predictor for train and test.

The mean predictor on the training inputs obeys the following integral equation

$$\langle f_{\text{train}}(t)\rangle = \int_0^t dt' K_d^L(t,t')\left(Y - \langle f_{\text{train}}(t')\rangle\right) \tag{7}$$

where the average on $\langle f_{\text{train}}(t)\rangle$ is over all possible trajectories of the parameters, encompassing both the randomness of the noise and the initial condition. The mean predictor on any test point $\mathbf{x}$ is given by

$$\langle f(t,\mathbf{x})\rangle = \int_0^t dt' k_d^L(t,t',\mathbf{x})^\top \left(Y - \langle f_{\text{train}}(t')\rangle\right) \tag{8}$$

The quantities $K_d^L(t,t')$,$k_d^L(t,t',\mathbf{x})$ appearing in Eq.7, are $P\times P$ matrix and $P$ vector, respectivly, such that $K^L_{d,\mu\nu}(t,t') = \mathcal{K}_d^L(t,t',\mathbf{x}_\mu,\mathbf{x}_\nu)$, $k^L_{d,\mu}(t,t',\mathbf{x}) = \mathcal{K}_d^L(t,t',\mathbf{x}_\mu,\mathbf{x})$. They are defined via a time-dependent Neural Dynamical Kernel (NDK) function [1], which is for any two inputs

$$\mathcal{K}_d^L(t,t',\mathbf{x},\mathbf{x}') = e^{-T\sigma^{-2}|t-t'|}\left\langle \nabla_\Theta f(t,\mathbf{x})\cdot\nabla_\Theta f(t',\mathbf{x}')\right\rangle_0 \tag{9}$$

Where $\langle\,\rangle_0$ denotes averaging over the time-dependent prior distribution, and is described in SI Eq.59. The NDK has closed-form expressions for some nonlinearities such as ReLU and error function, which are given in SI Sec.E (inspired by the static expressions for nonlinear kernels [7, 34]).

This set of equations has been shown to describe both the NTK theory in the limit of $T\to 0, t\sim O(1)$ and the NNGP theory in the limit $t\to\infty$. However, as previously discussed, in the lazy learning regime the leading-order of the representations is not affected by learning, and an analytical expression for the first-order correction for a general nonlinearity is still unknown.

In addition, characterizing feature learning dynamics in nonlinear non-lazy (mean-field) scaling regime, $\gamma = 1$, remains challenging and has yet to be fully solved for Langevin dynamics. Therefore, we focus on a simpler setting that enables analytical derivation of a broader range of relevant quantities.

## 4. Langevin Dynamics in Linear Networks

We describe the dynamics of a linear network with one hidden layer and a single output, learning with Langevin dynamics (Eq. 4). The dynamics of the vector $\mathbf{a}(t)$ and the matrix $\mathbf{W}(t)$ are given by

$$\frac{d}{dt}\mathbf{a}_t = P\left(\frac{1}{N^{1-\gamma}}\mathbf{W}_t\mathbf{q} - \frac{1}{N}\mathbf{W}_t\Sigma\mathbf{W}_t^\top\mathbf{a}_t\right) - T\sigma^{-2}\mathbf{a}_t + \xi_a(t) \tag{10}$$

$$\frac{d}{dt}\mathbf{W}_t = P\left(\frac{1}{N^{1-\gamma}}\mathbf{a}_t\mathbf{q}^\top - \frac{1}{N}\mathbf{a}_t\mathbf{a}_t^\top\mathbf{W}_t\Sigma\right) - T\sigma^{-2}\mathbf{W}_t + \xi_W(t) \tag{11}$$

Where $N_L \equiv N$. To account for the different normalizations, we rescale $dt$ by a factor $N^{2\gamma-1}$ as was previously suggested [5], and scale the temperature accordingly [32].

We define $\Sigma = \frac{1}{PN_0}\sum_{\mu=1}^{P}\mathbf{x}_\mu\mathbf{x}_\mu^\top \in \mathbb{R}^{N_0\times N_0}$, the data covariance matrix, and the input-output correlation vector $\mathbf{q} = \frac{1}{P\sqrt{N_0}}\sum_{\mu=1}^{P}\mathbf{x}_\mu y_\mu \in \mathbb{R}^{N_0}$. $\xi_a(t), \xi_W(t)$ are white noise terms similar to the ones described in Sec.2. Generalizing methods from [28], we express $\mathbf{W}_t$ in the eigenbasis of $\Sigma$. We construct an orthonormal basis out of the eigenvectors of $\Sigma$, $\sigma_{1\dots N_0}$, and use the decomposition $\mathbf{W}_t = \sum_{n=1}^{N_0}\mathbf{w}_n(t)\sigma_n^\top$. For Simplicity, here we assume that $|\mathbf{q}| = 1$, and that $\mathbf{q}$ is aligned with an eigenvector of $\Sigma$, such that $\sigma_q = \mathbf{q}$, with eigenvalue $\lambda_q$. The general case is treated in SI Sec.A. We note that $\mathbf{w}_q$ plays a special role in the dynamics, as it is the pair of the task vector $\mathbf{q}$ in the decomposition of $\mathbf{W}_t$.

We derive scalar equations by projecting the vectors onto one another $\mathcal{A}(t) = \frac{1}{N}\langle\mathbf{a}^2(t)\rangle$, $\mathcal{W}_{nm}(t) = \frac{1}{N}\langle\mathbf{w}_n(t)\cdot\mathbf{w}_m(t)\rangle$, $v_n(t) = \frac{1}{N^\gamma}\langle\mathbf{w}_n(t)\cdot\mathbf{a}(t)\rangle$. We note that $\mathcal{A}(t)$ and $\mathcal{W}_{nm}(t)$ are self-averaging and do not fluctuate, while $v_n(t)$ fluctuates in standard scaling ($\gamma = 1/2$) but not in mean-field scaling ($\gamma = 1$), similar to the predictor in each case. We can write the equations for all $O(N_0^2)$ variables (see SI Sec. A). However, under the simplified assumptions we stated above, $N \gg N_0$ and Gaussian initial conditions, only three variables participate in the dynamics: $\mathcal{A}(t)$, $\mathcal{W}_{qq}(t)$, and $v_q(t)$. Moreover, $\mathcal{A}(t)$ and $\mathcal{W}_{qq}(t)$ follow the same dynamics with identical initial conditions, allowing us to solve for just for one of them (see SI Sec. A for details). As a result, the system is reduced to two nonlinear scalar ODEs:

$$\frac{d}{dt}v_q(t) = 2P\mathcal{A}(t)\left(1-\lambda_q v_q(t)\right) - 2T\sigma^{-2}v_q(t) \tag{12}$$

$$\frac{d}{dt}\mathcal{A}(t) = 2\alpha N^{2\gamma-1}v_q(t)\left(1-\lambda_q v_q(t)\right) + 2T\left(1-\sigma^{-2}\mathcal{A}(t)\right) \tag{13}$$

With initial conditions $\mathcal{A}(0) = \sigma_0^2, v_q(0) = 0$.

### 4.1. The Predictor and Representations

The predictor on a general point $\mathbf{x}$ is dependent only on $v_q(t)$

$$\langle f(\mathbf{x},t)\rangle = v_q(t)\left(\frac{1}{PN_0}\sum_{\mu=1}^{P}(x_\mu\cdot x)y_\mu\right) = \lambda_q v_q(t) f_{eq} \tag{14}$$

Where $f_{eq}$ is the usual predictor given by the equilibrium solution in linear networks, $k_0^\top K_0^{-1}Y$ or $q^\top\Sigma^{-1}\mathbf{x}$, depending on the ratio between $P$ and $N_0$. Thus analysis of the dynamics of $v_q(t)$ is sufficient to understand the dynamics of the mean predictor.

In addition, we can look at the dynamics of the kernel as a means to understand the representations in the model:

$$K(t) = K_0 m(t,t) + \lambda_q^2 YY^\top(\mathcal{A}(t) - m(t,t)) \tag{15}$$

As was predicted before by equilibrium calculations [10, 19], there is a low rank learned component, and it is dependent upon the deviation $\mathcal{A}(t)$ from its prior dynamics. In the lazy regime, the change in $\mathcal{A}(t)$ due to learning is proportional to $P/N$, and thus small in the infinite width limit. However, in the non-lazy case, the deviation is controlled by $P$, and thus not small in general.

### 4.2. Lazy Regime

For $\gamma = 1/2$, the factor $N^{2\gamma-1}$ disappears. $\alpha = P/N \to 0$ in the infinite width limit, and thus the equation for $\mathcal{A}(t)$ decouples from $v_q(t)$. Solving it yields $\mathcal{A}(t) = m(t,t)$ Which is the usual prior for lazy networks. We can substitute $\mathcal{A}(t)$ in the equation for $v_q(t)$ and get a solution in terms of an integral (see SI Sec.D.2), which recovers the solution for linear NDK of Eqs.7,8. In the limit of $T \to 0$, we get the NTK solution $v_q(t) = 1 - \exp(-2\lambda_q \sigma_0^2 t)$.

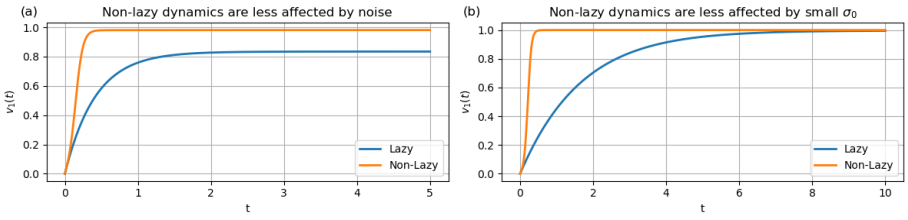


Figure 1: Lazy and Non-Lazy Dynamics: (a) Comparison between lazy and non-lazy dynamics for $T = 0.2$, $P = 100$, $\sigma = \sigma_0 = 1$, and $N = 10^4$. In the non-lazy network, the equilibrium depends on the ratio between $P$ and $T$, leading to only a small deviation from the zero-temperature equilibrium ($v_1 = 1$ in this case). In contrast, in the lazy network, the deviation is governed by the ratio between $\sigma_0$ and $T$, and is therefore more strongly influenced by the temperature. (b)$T = 0.001$, $P = 100$, $\sigma = 1, \sigma_0 = 0.3$. In non-lazy dynamics, only the initial slope is controlled by $\sigma_0$, while the time constant is related to $P$. In contrast, in the lazy regime, the time constant scales with $\sigma_0$, resulting in slower dynamics.

### 4.3. Non-Lazy Learning

For $\gamma = 1$, we get the equations

$$\frac{d}{dt} v_q(t) = 2P\mathcal{A}(t)\left(1 - \lambda_q v_q(t)\right) - 2T\sigma^{-2} v_q(t) \tag{16}$$

$$\frac{d}{dt}\mathcal{A}(t) = 2P v_q(t)\left(1 - \lambda_q v_q(t)\right) + 2T\left(1 - \sigma^{-2}\mathcal{A}(t)\right) \tag{17}$$

In general, this set of nonlinear ODEs needs to be solved numerically. However, for $T = 0$, the solution for $\mathcal{A}(v_q)$ yields $\mathcal{A}(v) = \sqrt{v_q^2 + \sigma_0^2}$, and thus at low $T$, we can conclude that the deviation of $\mathcal{A}(t)$ from the prior is $O(1)$, and the kernel has significant low rank feature learning component (see Eq.15). We note that $\sigma_0^2$ determines the timescale of convergence in the lazy case. However, in the non-lazy case, the time scale is related to $v_q$, where $\sigma_0$ only determines the initial slope. Thus, if $\sigma_0$ is small, the lazy dynamics would be slow, while the non-lazy dynamics would be only slightly affected as can be seen in Fig. 1a. In addition, if we consider a finite $T$, $v_q = 1$ is no longer the equilibrium solution as the two terms of Eq.16 need to be balanced, and the ratio between them is proportional to the ratio between $\mathcal{A}(t)$ and $T$. In the lazy case, the ratio between $\sigma^2$ and $T$ would determine the equilibrium solution, where in the nonlazy case, the ratio between $P$ and $T$ would determine the equilibrium. Thus, the non-lazy dynamics are less affected by finite noise.

## 5. Summary

This work introduces a dynamical theory of Langevin learning in wide neural networks, for both the lazy and non-lazy learning regimes. We extend current works on lazy nonlinear deep models, and utilize simpler linear models to study how noise, initialization, and network scaling affect both the predictor dynamics and kernel structure. We aim to extend the theory to study the temporal correlations in the non-lazy regime. In the lazy regime, analyzing these correlations at equilibrium has shed light on brain phenomena such as representational drift [1, 26]. Understanding how different regularization schemes affect such phenomena remains an open question.

## References

[1] Yehonatan Avidan, Qianyi Li, and Haim Sompolinsky. Unified theoretical framework for wide neural network learning dynamics. *Physical Review E*, 111(4):045310, 2025.

[2] Juhan Bae, Paul Vicol, Jeff Z HaoChen, and Roger B Grosse. Amortized proximal optimization. *Advances in Neural Information Processing Systems*, 35:8982–8997, 2022.

[3] Yasaman Bahri, Jonathan Kadmon, Jeffrey Pennington, Sam S Schoenholz, Jascha Sohl-Dickstein, and Surya Ganguli. Statistical mechanics of deep learning. *Annual Review of Condensed Matter Physics*, 11:501–528, 2020.

[4] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[5] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *arXiv preprint arXiv:2205.09653*, 2022.

[6] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

[7] Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. *Advances in neural information processing systems*, 22, 2009.

[8] William Coffey and Yu P Kalmykov. *The Langevin equation: with applications to stochastic problems in physics, chemistry and electrical engineering*, volume 27. World Scientific, 2012.

[9] Dmitriy Drusvyatskiy and Adrian S Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.

[10] Kirsten Fischer, Javed Lindner, David Dahmen, Zohar Ringel, Michael Krämer, and Moritz Helias. Critical feature learning in deep neural networks. *arXiv preprint arXiv:2405.10761*, 2024.

[11] Silvio Franz and Giorgio Parisi. Effective potential in glassy systems: theory and simulations. *Physica A: Statistical Mechanics and its Applications*, 261(3-4):317–339, 1998.

[12] Silvio Franz, Giorgio Parisi, and Miguel Angel Virasoro. The replica method on and off equilibrium. *Journal de Physique I*, 2(10):1869–1880, 1992.

[13] Marylou Gabrié, Andre Manoel, Clément Luneau, Nicolas Macris, Florent Krzakala, Lenka Zdeborová, et al. Entropy and mutual information in models of deep neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

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

[15] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[16] Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

[17] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

[18] Qianyi Li and Haim Sompolinsky. Statistical mechanics of deep linear neural networks: The backpropagating kernel renormalization. *Physical Review X*, 11(3):031059, 2021.

[19] Qianyi Li and Haim Sompolinsky. Globally gated deep linear networks. *arXiv preprint arXiv:2210.17449*, 2022.

[20] Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

[21] Gadi Naveh, Oded Ben David, Haim Sompolinsky, and Zohar Ringel. Predicting the outputs of finite deep neural networks trained with noisy gradients. *Physical Review E*, 104(6):064301, 2021.

[22] Radford M Neal. Priors for infinite networks (tech. rep. no. crg-tr-94-1). *University of Toronto*, 415, 1994.

[23] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

[24] Nicholas G. Polson, James G. Scott, and Brandon T. Willard. Proximal algorithms in statistics and machine learning. *arXiv preprint arXiv:1502.07944*, 2015.

[25] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[26] Michael E Rule, Timothy O'Leary, and Christopher D Harvey. Causes and consequences of representational drift. *Current opinion in neurobiology*, 58:141–147, 2019.

[27] Luca Saglietti and Lenka Zdeborová. Solvable model for inheriting the regularization through knowledge distillation. In *Mathematical and Scientific Machine Learning*, pages 809–846. PMLR, 2022.

[28] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. URL https://arxiv.org/abs/1312.6120.

[29] Haozhe Shan, Qianyi Li, and Haim Sompolinsky. Order parameters and phase transitions of continual learning in deep neural networks. *arXiv preprint arXiv:2407.10315*, 2024.

[30] Marc Teboulle. Convergence of proximal-like algorithms. *SIAM Journal on Optimization*, 7 (4):1069–1083, 1997.

[31] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[32] Alexander van Meegen and Haim Sompolinsky. Coding schemes in neural networks learning classification tasks. *arXiv preprint arXiv:2406.16689*, 2024.

[33] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[34] Christopher Williams. Computing with infinite networks. *Advances in neural information processing systems*, 9, 1996.

[35] Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR, 2020.

[36] Greg Yang and Edward J Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737. PMLR, 2021.

# Supplemental Information

## Appendix A. Langevin Dynamics in Linear Networks

In this section, we derive the ODEs for a general data covariance matrix, which the results in Sec.4 are a special case. In linear networks with one hidden layer, the predictor is given by

$$f_t^\mu = \frac{1}{N^\gamma}\left(\frac{\mathbf{W}\mathbf{x}_\mu}{\sqrt{N_0}}\right)^\top \mathbf{a}_t \tag{1}$$

Taking the gradient of the cost function (Eq.3) w.r.t. $\mathbf{a}$, $\mathbf{W}$, we get

$$\frac{d}{dt}\mathbf{a}_t = P\left(\frac{1}{N^{1-\gamma}}\mathbf{W}_t\mathbf{q} - \frac{1}{N}\mathbf{W}_t\Sigma\mathbf{W}_t^\top\mathbf{a}_t\right) - T\sigma^{-2}\mathbf{a}_t + \xi_a(t) \tag{2}$$

$$\frac{d}{dt}\mathbf{W}_t = P\left(\frac{1}{N^{1-\gamma}}\mathbf{a}_t\mathbf{q}^\top - \frac{1}{N}\mathbf{a}_t\mathbf{a}_t^\top\mathbf{W}_t\Sigma\right) - T\sigma^{-2}\mathbf{W}_t + \xi_W(t) \tag{3}$$

The dynamics of $\mathbf{a}$ and $\mathbf{w}_n$

Where

$$\Sigma = \frac{1}{PN_0}\sum_{\mu=1}^{P}\mathbf{x}_\mu\mathbf{x}_\mu^\top, \quad \mathbf{q} = \frac{1}{P\sqrt{N_0}}\sum_{\mu=1}^{P}y_\mu\mathbf{x}_\mu \tag{4}$$

$\Sigma$ is a real symmetric matrix and thus can be diagonalized with real eigenvalues and orthonormal eigenvectors. We denote its eigenvectors $\sigma_{1\ldots N_0}$, and eigenvalues $\lambda_{1\ldots rank(\Sigma)}$, and the rest are zero. We a scalar overlap $\kappa_n = \sigma_n \cdot \mathbf{q}$

We can span the $N_0$ dimension of $\mathbf{W}$ on the eigenvectors

$$\mathbf{W}_t = \sum_{n=1}^{N_0}\mathbf{w}_n(t)\,\sigma_n^\top \tag{5}$$

The dynamics of $\mathbf{a}$, $\mathbf{w}_n$ are

$$\frac{d}{dt}\mathbf{w}_n(t) = \frac{P}{N}\left(N^\gamma\kappa_n - \lambda_n\left(\mathbf{a}_t\cdot\mathbf{w}_n(t)\right)\right)\mathbf{a}_t - T\sigma^{-2}\mathbf{w}_n(t) + \xi_{\mathbf{w}_n}(t) \tag{6}$$

$$\frac{d}{dt}\mathbf{a}_t = \frac{P}{N}\left(N^\gamma\sum_{n=1}^{rank(\Sigma)}\kappa_n\mathbf{w}_n(t) - \sum_{n=1}^{rank(\Sigma)}\lambda_n\left(\mathbf{a}_t\cdot\mathbf{w}_n(t)\right)\mathbf{w}_n(t)\right) - T\sigma^{-2}\mathbf{a}_t + \xi_a(t) \tag{7}$$

where $\xi_{\mathbf{w}_n}(t) = \sigma_n\cdot\xi_W(t)$, and because $\sigma_n$ are orthonormal, follow white noise statistics (Eq.4).

We define new scalar variables, $\mathcal{A}(t) = \frac{1}{N}\langle\mathbf{a}^2(t)\rangle, \mathcal{W}_{nm}(t) = \frac{1}{N}\langle\mathbf{w}_n(t)\cdot\mathbf{w}_m(t)\rangle, v_n(t) = \frac{1}{N^\gamma}\langle\mathbf{w}_n(t)\cdot\mathbf{a}_t\rangle$, The equations for these variables are

$$\frac{d}{dt}\mathcal{A}(t) = 2\alpha N^{2\gamma-1}\sum_{n=1}^{rank(\Sigma)}v_n(t)\left(\kappa_n - \lambda_n v_n(t)\right) + 2T\left(1-\sigma^{-2}\mathcal{A}(t)\right) \tag{8}$$

$$\frac{d}{dt}\mathcal{W}_{nm}(t) = \alpha N^{2\gamma-1}\left(v_m(t)\left(\kappa_n - \lambda_n v_n(t)\right) + v_n(t)\left(\kappa_m - \lambda_m v_m(t)\right)\right) + 2T\left(\delta_{nm} - \sigma^{-2}\mathcal{W}_{nm}(t)\right) \tag{9}$$

$$\frac{d}{dt}v_n(t) = \left(\sum_{m=1}^{rank(\Sigma)}\left(\mathcal{W}_{nm}(t) + \delta_{nm}\mathcal{A}(t)\right)\left(\kappa_m - \lambda_m v_m(t)\right)\right) - 2T\sigma^{-2}v_n(t) \tag{10}$$

Where we averages over the noise.

We consider the initial condition in the infinite width limit $N \gg N_0$

$$\mathcal{W}_{nm}(0) = \delta_{nm}\sigma_0^2, \quad \mathcal{A}(0) = \sigma_0^2 \quad v_n(0) = 0 \tag{11}$$

We mention specifically the case described in Sec.4. In this scenario, $\kappa_n = \delta_{n,q}$. We can see that any $v_n(t)$ with $\kappa_n = 0$ won't participate in the dynamics, as there is a trivial solution of $v_n(t) = 0$ for all times. Thus, the only $v_n(t)$ that has any dynamics is $v_q(t)$. For the same reasons, only $\mathcal{W}_{qq}(t)$ has dynamics, the other diagonal $\mathcal{W}_{qq}(t)$ depends only on the prior, and the non diagonal are zero.

We get the equations:

$$\frac{d}{dt}v_q(t) = \left(\mathcal{A}(t) + \mathcal{W}_{qq}(t)\right)\left(1 - v_q(t)\right) - 2T\sigma^{-2}v_q(t) \tag{12}$$

$$\frac{d}{dt}\mathcal{W}_{qq}(t) = 2\alpha N^{2\gamma-1}v_q(t)\left(1 - v_q(t)\right) + 2T\left(1 - \sigma^{-2}\mathcal{W}_{qq}(t)\right) \tag{13}$$

$$\frac{d}{dt}\mathcal{A}(t) = 2\alpha N^{2\gamma-1}v_q(t)\left(1 - v_q(t)\right) + 2T\left(1 - \sigma^{-2}\mathcal{A}(t)\right) \tag{14}$$

But since the equations for $\mathcal{W}_{qq}(t)$ and $\mathcal{A}(t)$ are the same with the same initial condition, we can infer that $\mathcal{A}(t) = \mathcal{W}_{qq}(t)$, and get Eqs. 12, 13 from the main text.

## Appendix B. Markov Proximal Learning

We introduce a Markov Proximal Learning (MPL) framework for learning dynamics in fully connected Deep Neural Networks (DNNs). This method allows us to construct a dynamical mean field theory for Langevin dynamics in the infinite width limit, and is a novel way to discritize Langevin dynamics and formulate out-of-equilibrium statistical mechanics. We formally write down the moment-generating function (MGF) of the predictor. We then use the well-known replica method in statistical physics [12, 20], which has also been shown to be a powerful tool for deriving analytical results for learning in NNs [3, 6, 13, 14, 27]. We analytically calculate the MGF after averaging over the posterior distribution of the network weights in the infinite width limit, which enables us to compute statistics of the predictor.

### B.1. Definition of Markov Proximal Learning

We consider the network learning dynamics as a Markov proximal process, which is a generalized version of the *deterministic* proximal algorithm ([23, 24]). Deterministic proximal algorithm with $L_2$ regularization is a sequential update rule defined as

$$\Theta_t\left(\Theta_{t-1}, \mathcal{D}\right) = \arg\min_{\Theta}\left(E\left(\Theta|\mathcal{D}\right) + \frac{\lambda}{2}\left|\Theta - \Theta_{t-1}\right|^2\right) \tag{15}$$

where $\lambda$ is a parameter determining the strength of the proximity constraint. This algorithm has been proven to converge to the global minimum for convex cost functions [9, 30], and many optimization algorithms widely used in machine learning can be seen as its approximations [2, 4, 25]. We define a stochastic extension of proximal learning, the Markov proximal learning. This method was also inspired by continual learning methods [29] and Franz-Parisi potential [11]. The process is characterized by the following transition matrix

$$\mathcal{T}\left(\Theta_t|\Theta_{t-1}\right)=\frac{1}{Z\left(\Theta_{t-1}\right)}\exp\left(-\frac{1}{2}\beta\left(E\left(\Theta_t\right)+\frac{\lambda}{2}\left|\Theta_t-\Theta_{t-1}\right|^2\right)\right) \tag{16}$$

where $Z\left(\Theta_{t-1}\right)$ is the single-time partition function, which imposes normalization throughout the Markov process, $Z\left(\Theta_{t-1}\right)=\int d\Theta'\exp\left(-\frac{1}{2}\beta\left(E\left(\Theta'\right)+\frac{\lambda}{2}\left|\Theta'-\Theta_{t-1}\right|^2\right)\right)$ $\beta$ is an inverse temperature parameter characterizing the level of 'uncertainty' and $\beta\to\infty$ limit recovers the deterministic proximal algorithm. We note that in the large $\lambda$ limit, the difference between $\Theta_t$ and $\Theta_{t-1}$ is infinitesimal, and $\Theta_t$ becomes a smooth function of continuous time, where the time variable is the discrete time divided by $\lambda$.

The joint probability of the parameters is given by $(\Theta_0,\Theta_1,...,\Theta_t)$.

$$P\left(\Theta_0,\Theta_1,...,\Theta_t\right)=\left[\prod_{\tau=1}^{t}\mathcal{T}\left(\Theta_\tau|\Theta_{\tau-1}\right)\right]P\left(\Theta_0\right) \tag{17}$$

where $P\left(\Theta_0\right)$ is the distribution of the initial condition of the parameters.

### B.2. Large $\lambda$ Limit and Langevin dynamics:

We prove that in the limit of large $\lambda$ and differentiable cost function this algorithm is equivalent to Langevin dynamics. We define $\delta\Theta_t=\Theta_t-\Theta_{t-1}$ . In the limit of large $\lambda$, we can expand the transition matrix around $\delta\Theta_t=0$:

$$\mathcal{T}\left(\delta\Theta_t|\Theta_{t-1}\right)\approx\left(\frac{\lambda\beta}{4\pi}\right)^{\frac{d}{2}}\exp\left[-\frac{\lambda\beta}{4}\left|\delta\Theta_t+\frac{1}{\lambda}\nabla E\left(\Theta_{t-1}\right)\right|^2\right] \tag{18}$$

$\delta\Theta_t|\Theta_{t-1}$ is a Gaussian random variable with the statistics:

$$\left\langle\delta\Theta_t|\Theta_{t-1}\right\rangle=-\frac{1}{\lambda}\nabla_\Theta E\left(\Theta_{t-1}\right) \tag{19}$$

$$\text{var}\left(\delta\Theta_t\delta\Theta_{t'}^\top|\Theta_{t-1}\right)=\frac{2}{\lambda\beta}\delta_{t,t'}I \tag{20}$$

which is equivalent to Langevin dynamics in Itô discretization:

$$\delta\Theta_t=\left(-\nabla_\Theta E\left(\Theta_{t-1}\right)+\eta_{t-1}\right)dt \tag{21}$$

with

$$\left\langle\eta_t\eta_{t'}^\top\right\rangle=\frac{2T}{dt}\delta_{t,t'}I,\left\langle\eta_t\right\rangle=0 \tag{22}$$

where $\frac{1}{\lambda}=dt$,$\beta=\frac{1}{T}$.

## Appendix C. The Statistics of the Predictor

### C.1. Replica Calculation of the Moment-Generating Function of the Predictor

The transition density can be written using the replica method, where $Z^{-1}\left(\Theta_{t-1}\right)=\lim_{n\to 0} Z^{n-1}\left(\Theta_{t-1}\right)$,:

$$\begin{aligned}\mathcal{T}\left(\Theta_t|\Theta_{t-1}\right) &= \lim_{n\to 0} Z^{n-1}\left(\Theta_{t-1}\right)\exp\left(-\frac{1}{2}\beta\left(E\left(\Theta_t\right)+\frac{\lambda}{2}\left|\Theta_t-\Theta_{t-1}\right|^2\right)\right) \\ &= \lim_{n\to 0}\prod_{\alpha=1}^{n-1}\int d\Theta_t^{\alpha}\exp\left(-\frac{\beta}{2}\left(\sum_{\alpha=1}^{n}E\left(\Theta_t^{\alpha}\right)+\frac{\lambda}{2}\sum_{\alpha=1}^{n}\left|\Theta_t^{\alpha}-\Theta_{t-1}^{n}\right|^2\right)\right)\end{aligned} \tag{23}$$

Here $\alpha=1,\cdots,n-1$ are the 'replicated copies' of the physical variable $\{\Theta_{\tau}^{n}\}_{\tau=1,\cdots,t}$. To calculate the statistics of the dynamical process, we consider the MGF for arbitrary functions of the trajectory $g(\{\Theta_{\tau}^{n}\}_{\tau=0,\cdots t})$

$$\begin{aligned}\mathcal{M}\left[\ell\right] &= \left\langle\exp\left(\sum_{t=1}^{\infty}\ell_t g\left(\{\Theta_{\tau}^{n}\}_{\tau=0,\ldots,t}\right)\right)\right\rangle_{\Theta} \\ &= \prod_{\tau=0}^{\infty}\int d\Theta_{\tau}\left[\prod_{\tau=1}^{\infty}\mathcal{T}\left(\Theta_{\tau}|\Theta_{\tau-1}\right)\right]P\left(\Theta_0\right)\exp\left(\sum_{t=1}^{\infty}\ell_t g\left(\{\Theta_{\tau}^{n}\}_{\tau=0,\ldots t}\right)\right) \\ &= \lim_{n\to 0}\prod_{\alpha=1}^{n}\prod_{\tau=1}^{\infty}\int d\Theta_t^{\alpha}\int d\Theta_0^{n}P\left(\Theta_0^{n}\right) \\ &\exp\left(-\frac{\beta}{2}\sum_{\tau=1}^{\infty}\left(\sum_{\alpha=1}^{n}E\left(\Theta_{\tau}^{\alpha}\right)+\frac{\lambda}{2}\sum_{\alpha=1}^{n}\left|\Theta_{\tau}^{\alpha}-\Theta_{\tau-1}^{n}\right|^2\right)+\sum_{t=1}^{\infty}\ell_t g\left(\{\Theta_{\tau}^{n}\}_{\tau=0,\cdots t}\right)\right)\end{aligned} \tag{24}$$

We apply this formalism to the supervised learning cost function introduced in Eq.3 in the main text.

$$E\left(\Theta_t|\mathcal{D}\right)=\frac{1}{2}\sum_{\mu=1}^{P}\left(f\left(\mathbf{x}^{\mu},\Theta_t\right)-y^{\mu}\right)^2+\frac{T}{2\sigma^2}\left|\Theta_t\right|^2 \tag{25}$$

and the predictor statistics at time $t$, $g(\{\Theta_{\tau}^{n}\}_{\tau=0,\cdots t})=f\left(\mathbf{x},\Theta_t^{n}\right)$ ,yielding

$$\mathcal{M}\left[\ell\right]=\lim_{n\to 0}\prod_{\alpha=1}^{n}\prod_{\tau=1}^{\infty}\int d\Theta_{\tau}^{\alpha}\int d\Theta_0\exp\left(-\frac{\beta}{4}\sum_{\tau=1}^{\infty}\sum_{\alpha=1}^{n}\left(f_{\text{train}}\left(\Theta_{\tau}^{\alpha}\right)-Y\right)^2+\sum_{t=1}^{\infty}\sum_{\mathbf{x}}\ell_{t,\mathbf{x}}f\left(\mathbf{x},\Theta_t^{n}\right)-S_0\left[\Theta\right]\right) \tag{26}$$

$$S_0\left[\Theta\right]=\frac{1}{4}\sum_{\tau=1}^{\infty}\sum_{\alpha=1}^{n}\left(\sigma^{-2}\left|\Theta_{\tau}^{\alpha}\right|^2+\lambda\beta\left|\Theta_{\tau}^{\alpha}-\Theta_{\tau-1}^{n}\right|^2\right)+\frac{1}{2}\sigma_0^{-2}\left|\Theta_0^{n}\right|^2 \tag{27}$$

Where we define $f_{\text{train}}\left(\Theta_{\tau}^{\alpha}\right)\equiv\left[f\left(\mathbf{x}^1,\Theta_{\tau}^{\alpha}\right),\cdots,f\left(\mathbf{x}^P,\Theta_{\tau}^{\alpha}\right)\right]^T\in\mathbb{R}^P$ a vector contains the predictor on the training dataset, and $Y\in\mathbb{R}^P$ such that $Y^{\mu}=y^{\mu}$. $S_0\left[\Theta\right]$ denote the Gaussian prior on the parameters including the hidden layer weights and the readout weights.

To perform the integration over $\mathbf{a}_\tau^\alpha$, we use Hubbard-Stratonovich (H.S.) transformation and introduce a new vector field $v_\tau^\alpha \in \mathbb{R}^P$

$$\mathcal{M}[\ell] = \lim_{n\to 0} \prod_{\alpha=1}^{n} \prod_{\tau=1}^{\infty} \int d\Theta_\tau^\alpha \int dv_\tau^\alpha \int d\Theta_0 \tag{28}$$
$$\exp\left( -\frac{i\beta}{2} \sum_{\tau=1}^{\infty} \sum_{\alpha=1}^{n} \left( \frac{1}{\sqrt{N_L}} f_{\text{train}}\left(\Theta_\tau^\alpha\right) - Y \right)^\top v_\tau^\alpha - \frac{\beta}{4} \sum_{\tau=1}^{\infty} \sum_{\alpha=1}^{n} |v_\tau^\alpha|^2 + \sum_{t=1}^{\infty} \sum_{\mathbf{x}} \ell_{t,\mathbf{x}} f\left(\mathbf{x}, \Theta_t^n\right) - S_0[\Theta] \right)$$

**Averaging over the readout weights:**

We integrate over $\mathbf{a}_\tau^\alpha$. For convenience, we denote the set of all hidden layer weights collectively as $\mathbf{W}_t = \left\{ \mathbf{W}_t^{\ell=1}, \ldots, \mathbf{W}_t^L \right\}$, similar to the main text.

$$\mathcal{M}[\ell] = \lim_{n\to 0} \prod_{\tau=1}^{\infty} \prod_{\alpha=1}^{n} \int dv_\tau^\alpha \int d\mathbf{W}_\tau^\alpha \exp\left( -S[v, \mathbf{W}] - Q[\ell, v, \mathbf{W}] - S_0[\mathbf{W}] \right) \tag{29}$$

$$S[v, \mathbf{W}] = \frac{\beta}{4} \left( \sum_{\alpha,\beta=1}^{n} \sum_{\tau=1}^{\infty} \frac{\beta}{2} v_\tau^{\alpha\top} m_{\tau,\tau'}^{\alpha\beta} K_{\tau,\tau'}^{L,\alpha\beta}\left(\mathbf{W}_\tau^\alpha\right) v_{\tau'}^\beta + \sum_{\alpha=1}^{n} \sum_{\tau=1}^{\infty} \left(v_\tau^\alpha - 2iY\right)^\top v_\tau^\alpha \right) \tag{30}$$

and the source term action

$$Q[\ell, v, \mathbf{W}] = i\frac{\beta}{2} \sum_{\alpha=1}^{n} \sum_{t,\tau=1}^{\infty} \sum_{\mathbf{x}} v_\tau^{\alpha\top} m_{t,\tau}^{\alpha n} k_{t,\tau}^{L,\alpha n}\left(\mathbf{W}_\tau^\alpha, \mathbf{x}\right) \ell_{t,\mathbf{x}} - \frac{1}{2} \sum_{t,t'=1}^{\infty} \sum_{\mathbf{x},\mathbf{x}'} m_{t,t'}^{nn} K_{t,t'}^{L,nn}\left(\mathbf{W}_\tau^n, \mathbf{x}, \mathbf{x}\right) \ell_{t,\mathbf{x}} \ell_{t',\mathbf{x}'} \tag{31}$$

Where $m_{\tau,\tau'}^{\alpha\beta}$ is a scalar function independent of the data, and represents the averaging w.r.t. to the replica dependent prior $S_0[\Theta]$, such that $\left\langle \left(\Theta_\tau^\alpha\right)_i \left(\Theta_{\tau'}^\beta\right)_j \right\rangle_{S_0} = \delta_{ij} m_{\tau,\tau'}^{\alpha\beta}$

$$m_{\tau,\tau'}^{\alpha\beta} = \begin{cases} m_{\tau,\tau'}^1 = \tilde{\sigma}^2 \left( \tilde{\lambda}^{|\tau-\tau'|} + \gamma \tilde{\lambda}^{\tau+\tau'} \right) & \{\alpha = \beta, \tau = \tau'\} \cup \{\alpha = n, \tau < \tau'\} \cup \{\beta = n, \tau > \tau'\} \\ m_{\tau,\tau'}^0 = \tilde{\sigma}^2 \left( \tilde{\lambda}^2 \tilde{\lambda}^{|\tau-\tau'|} + \gamma \tilde{\lambda}^{\tau+\tau'} \right) & otherwise \end{cases} \tag{32}$$

Where we have defined new functions of the parameters for convenience,

$$\tilde{\lambda} = \frac{\lambda}{\lambda + T\sigma^{-2}}, \tilde{\sigma}^2 = \sigma^2 \frac{\lambda + T\sigma^{-2}}{\lambda + \frac{1}{2}T\sigma^{-2}}, \gamma = \frac{\sigma_0^2}{\tilde{\sigma}^2} - 1 \tag{33}$$

The time-dependent and replica-dependent kernels $K_{\tau,\tau'}^{L,\alpha\beta} \in \mathbb{R}^{P\times P}, k_{\tau,\tau'}^{L,\alpha\beta}(\mathbf{x}) \in \mathbb{R}^P, K_{\tau,\tau'}^{L,\alpha\beta}(\mathbf{x}, \mathbf{x})$ defined as:

$$\mathcal{K}_{\tau,\tau'}^{L,\alpha\beta}\left(\mathbf{x}, \mathbf{x}'\right) = \frac{1}{N_L} \left( \mathbf{x}_\tau^L\left(\mathbf{x}, \mathbf{W}_\tau^\alpha\right) \cdot \mathbf{x}_{\tau'}^L\left(\mathbf{x}', \mathbf{W}_{\tau'}^\beta\right) \right) \tag{34}$$

And $K_{\tau,\tau'}^{L,\alpha\beta} \in \mathbb{R}^{P\times P}, k_{\tau,\tau'}^{L,\alpha\beta}(\mathbf{x}) \in \mathbb{R}^P$ are given by applying the kernel function on the training data and test data, respectively.

**Averaging over the hidden layer weights:**

In the infinite width limit, the statistics of $\mathbf{W}_\tau^\alpha$ is dominated by its Gaussian prior (Eq.27) with zero mean and covariance $\langle \mathbf{W}_\tau^\alpha \mathbf{W}_{\tau'}^{\beta\top} \rangle = m_{\tau,\tau'}^{\alpha\beta} I$ .Thus the averaged kernel function $K_{\tau,\tau'}^{\alpha\beta}$ (Eq.34) over the prior yields two kinds of statistics for a given pair of times $\{\tau, \tau'\}$, which we denote as $\mathcal{K}_{\tau,\tau'}^{1,L}(\mathbf{x},\mathbf{x}')$, and $\mathcal{K}_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}')$ :

$$\mathcal{K}_{\tau,\tau'}^{\alpha\beta} = \begin{cases} \mathcal{K}_{\tau,\tau'}^{1} & \{\alpha=\beta,\tau=\tau'\}\cup\{\alpha=n,\tau<\tau'\}\cup\{\beta=n,\tau>\tau'\} \\ \mathcal{K}_{\tau,\tau'}^{0} & otherwise \end{cases} \tag{35}$$

And they obey the iterative relations:

$$\mathcal{K}_{\tau,\tau'}^{1,L}(\mathbf{x},\mathbf{x}') = F\left(m_{\tau,\tau}^{1}\mathcal{K}_{\tau,\tau}^{1,L-1}(\mathbf{x},\mathbf{x}), m_{\tau',\tau'}^{1}\mathcal{K}_{\tau',\tau'}^{1,L-1}(\mathbf{x}',\mathbf{x}'), m_{\tau,\tau'}^{1}\mathcal{K}_{\tau,\tau'}^{1,L-1}(\mathbf{x},\mathbf{x}')\right) \tag{36}$$

$$\mathcal{K}_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}') = F\left(m_{\tau,\tau}^{1}\mathcal{K}_{\tau,\tau}^{1,L-1}(\mathbf{x},\mathbf{x}), m_{\tau',\tau'}^{1}\mathcal{K}_{\tau',\tau'}^{1,L-1}(\mathbf{x}',\mathbf{x}'), m_{\tau,\tau'}^{0}\mathcal{K}_{\tau,\tau'}^{0,L-1}(\mathbf{x},\mathbf{x}')\right) \tag{37}$$

$$\mathcal{K}^{1,L=0}(\mathbf{x},\mathbf{x}') = \mathcal{K}^{0,L=0}(\mathbf{x},\mathbf{x}') = \mathcal{K}^{in}(\mathbf{x},\mathbf{x}') \tag{38}$$

$$\mathcal{K}_{in}(\mathbf{x},\mathbf{x}') = \frac{1}{N_0}\sum_{i=1}^{N_0}\mathbf{x}_i\mathbf{x}_i' \tag{39}$$

where $F\left(\langle z^2\rangle, \langle z'^2\rangle, \langle zz'\rangle\right)$ is a nonlinear function of the variances of two Gaussian variables $z$ and $z'$ and their covariance, whose form depends on the nonlinearity of the network [7]. As we see in Eqs.36,37 these variances and covariances depend on the kernel functions of the previous layer and on the replica-dependent prior statistics represented by $m_{\tau,\tau'}^{1,0}$.

The MGF can be written as a function of the statistics of one of these kernels, and their difference, which we will denote as $\Delta_{\tau,\tau'}^{L}(\mathbf{x},\mathbf{x}') = \frac{\lambda\beta}{2}\left(\mathcal{K}_{\tau,\tau'}^{1,L}(\mathbf{x},\mathbf{x}') - \mathcal{K}_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}')\right)$. It is useful to define a new kernel, the discrete neural dynamical kernel $K_{\tau,\tau'}^{d,L} = \lim_{n\to 0}\frac{\lambda\beta}{2}\sum_{\alpha=1}^{n} m_{\tau,\tau'}^{n\beta}K_{\tau,\tau'}^{n\beta,L}$, which controls the dynamics of the mean predictor. It has a simple expression in terms of the kernel $\mathcal{K}_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}')$ and the kernel difference $\Delta_{\tau,\tau'}^{L}$.

$$\mathcal{K}_{\tau,\tau'}^{d,L}(\mathbf{x},\mathbf{x}') = \begin{cases} 0 & \tau\le\tau' \\ m_{\tau,\tau'}^{1}\Delta_{\tau,\tau'}^{L}(\mathbf{x},\mathbf{x}') + \tilde{\lambda}^{|\tau-\tau'|+1}\mathcal{K}_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}') & \tau>\tau' \end{cases} \tag{40}$$

We integrate over the replicated hidden layers variables $\mathbf{W}_\tau^\alpha$, which replaces the $\mathbf{W}_\tau^\alpha$ dependent kernels with the averaged kernels. We thus get an MGF that depends only of the $v_\tau^\alpha$ variables

$$\mathcal{M}[\ell] = \lim_{n\to 0}\prod_{\alpha=1}^{n}\prod_{\tau=1}^{\infty}\int dv_\tau^\alpha \exp\left(-S[v] - Q[\ell,v]\right) \tag{41}$$

$$S[v] = \frac{\beta}{4}\sum_{\tau=1}^{\infty}\left(\frac{\beta}{2}\sum_{\alpha,\beta=1}^{n}\sum_{\tau'=1}^{\infty} v_\tau^{\alpha\top}m_{\tau,\tau'}^{0}K_{\tau,\tau'}^{0}v_{\tau'}^{\beta} + \frac{2}{\lambda}\sum_{\alpha=1}^{n}\sum_{\tau'=1}^{t-1} v_\tau^{\alpha\top}K_{\tau,\tau'}^{d}v_{\tau'}^{n}\right. \\ \left. + \frac{1}{\lambda}\sum_{\alpha=1}^{n} v_\tau^{\alpha\top}K_{\tau,\tau}^{d}v_\tau^{\alpha} + \sum_{\alpha=1}^{n} v_\tau^{\alpha\top}\left(v_\tau^{\alpha} - 2iY\right)\right) \tag{42}$$

$$Q\left[\ell,v\right]=\frac{i\beta}{2}\sum_{\beta=1}^{n}\sum_{t,\tau'=1}^{\infty}\sum_{\mathbf{x}}\ell_{t,\mathbf{x}}m_{t,\tau'}^{0}k_{t,\tau'}^{0\top}\left(\mathbf{x}\right)v_{\tau'}^{\beta}+\frac{i}{\lambda}\sum_{t,\tau'=1}^{t}\sum_{\mathbf{x}}\ell_{t,\mathbf{x}}k_{t,\tau'}^{d\top}\left(\mathbf{x}\right)v_{\tau'}^{n} \tag{43}$$
$$+\frac{i}{\lambda}\sum_{\beta=1}^{n}\sum_{t=1}^{\infty}\sum_{\tau'=t+1}^{\infty}\sum_{\mathbf{x}}\ell_{t,\mathbf{x}}k_{t,\tau'}^{d\top}\left(\mathbf{x}\right)v_{\tau'}^{\beta}-\sum_{t=1}^{\infty}\sum_{\mathbf{x},\mathbf{x}'}\frac{1}{2}m_{t,t'}^{1}\ell_{t,\mathbf{x}}\ell_{t',\mathbf{x}'}\mathcal{K}_{t,t'}^{1}\left(\mathbf{x},\mathbf{x}'\right)$$

### C.2. Integrate Out Replicated Variables $v_\tau^\alpha$

We define a new variable $u_\tau=\frac{\lambda\beta}{2}\sum_{\alpha=1}^{n}v_\tau^\alpha$, and integrate out $v_\tau^{\alpha\neq n}$. We obtain a simpler expression of the MGF which is no longer replica dependent (after taking the limit $n\to 0$).

$$\mathcal{M}\left[\ell\right]=\prod_{\tau=1}^{\infty}\int dv_\tau\int du_\tau\exp\left(-S\left[v,u\right]-Q\left[\ell,v,u\right]\right) \tag{44}$$

$$S\left[v,u\right]=\frac{1}{2\lambda^2}\sum_{\tau,\tau'=1}^{\infty}u_\tau^\top\left(m_{\tau,\tau'}^{0}K_{\tau,\tau'}^{0}-\frac{2}{\beta}\delta_{\tau,\tau'}\left(I+\frac{1}{\lambda}K_{\tau,\tau}^{d}\right)\right)u_{\tau'} \tag{45}$$
$$+\frac{1}{\lambda}\sum_{\tau=1}^{\infty}\left(\frac{1}{\lambda}\sum_{\tau'=1}^{\tau-1}K_{\tau,\tau'}^{d}v_{\tau'}+\left(I+\frac{1}{\lambda}K_{\tau,\tau}^{d}\right)v_\tau-iY\right)^\top u_\tau$$

$$Q\left[\ell,v,u\right]=\frac{i}{\lambda}\sum_{t=1}^{\infty}\sum_{\mathbf{x}}\ell_{t,\mathbf{x}}\left(\sum_{\tau'=1}^{\infty}m_{t,\tau'}^{0}k_{t,\tau'}^{0\top}u_{\tau'}+\sum_{\tau'=1}^{t}k_{t,\tau'}^{d\top}v_{\tau'}+\frac{2}{\lambda\beta}\sum_{\tau'=t+1}^{\infty}k_{t,\tau'}^{d\top}u_{\tau'}\right) \tag{46}$$
$$-\sum_{t,t'=1}^{\infty}\sum_{\mathbf{x},\mathbf{x}'}\frac{1}{2}\ell_{t,\mathbf{x}}\ell_{t',\mathbf{x}'}m_{t,t'}^{1}k_{t,t'}^{1}\left(\mathbf{x},\mathbf{x}\right)$$

### C.3. Detailed Calculation of the Mean Predictor

To derive the mean predictor we take the derivative of the MGF w.r.t. $\ell_{t,\mathbf{x}}$:

$$\left\langle f\left(t,\mathbf{x}\right)\right\rangle=\left.\frac{\partial\mathcal{M}\left[\ell\right]}{\partial\ell_{t,\mathbf{x}}}\right|_{\ell_{t,\mathbf{x}}=0} \tag{47}$$

which yields

$$\left\langle f\left(t,\mathbf{x}\right)\right\rangle=\frac{1}{\lambda}\sum_{t'=1}^{t}k_{t,t'}^{d,L\top}\left(\mathbf{x}\right)\left\langle -iv_{t'}\right\rangle \tag{48}$$

Furthermore, from the H.S. transformation in Eq.28, we can relate $\langle v_\tau\rangle$ to the mean predictor on the training data $f_{\text{train}}\left(t\right)$

$$iv_t=f_{\text{train}}\left(t\right)-Y \tag{49}$$

For all moments of $f_{\text{train}}\left(t\right)$. On the other hand we can get the statistics of $iv_t$ from the MGF in Eq.44.

$$\left\langle f_{\text{train}}(t)\right\rangle=\left(I\lambda+K_{t,t}^{d,L}\right)^{-1}\sum_{t'=1}^{t-1}K_{t,t'}^{d,L}\left(Y-\left\langle f_{\text{train}}(t')\right\rangle\right) \tag{50}$$

$$\langle f\left(t,\mathbf{x}\right)\rangle=\frac{1}{\lambda}\sum_{t'=1}^{t}k_{t,t'}^{d,L\top}\left(\mathbf{x}\right)\left(Y-\langle\left(f_{\text{train}}\right)_{t'}\rangle\right) \tag{51}$$

where $K_{t,t'}^{d,L}$ is a $P\times P$ dimensional kernel matrix defined as $\mathcal{K}_{\mu\nu,t,t'}^{d,L}=K_{t,t'}^{d,L}\left(\mathbf{x}^{\mu},\mathbf{x}^{\nu}\right)$. Now we can compute $\langle f\left(\mathbf{x},\Theta_{t}\right)\rangle$ iteratively by combining Eqs.50,51.

### C.4. Large $\lambda$ Limit

All the results so far hold for any $T$ and $\lambda$. Now, we consider the limit where the Markov proximal learning algorithm is equivalent to Langevin dynamics in order to get expressions that are relevant to a continuous time gradient descent. We consider $\lambda\to\infty$ and $t_{discrete}\sim O\left(\lambda\right)$, and thus define a new continues time $t=t_{discrete}/\lambda\sim O\left(1\right)$ . In this limit, the parameters defined in Eq.33 becomes

$$\tilde{\lambda}^{t_{discrete}}=e^{-T\sigma^{-2}t},\tilde{\sigma}^{2}=\sigma^{2},\gamma=\frac{\sigma_{0}^{2}}{\sigma^{2}}-1 \tag{52}$$

Taking the limit of large $\lambda$ limit of Eq.44 is straightforward, and yields

$$\mathcal{M}\left[\ell\right]=\int Dv\int Du\exp\left(-S\left[v,u\right]-Q\left[\ell,v,u\right]\right) \tag{53}$$

Where

$$\begin{aligned}S\left[v,u\right]=&\frac{1}{2}\int_{0}^{\infty}dt\int_{0}^{\infty}dt'm\left(t,t'\right)u^{\top}\left(t\right)K^{L}\left(t,t'\right)u\left(t'\right)\\&+\int_{0}^{\infty}dt\left(\int_{0}^{t}dt'K_{d}^{L}\left(t,t'\right)v\left(t'\right)+v\left(t\right)-iY\right)^{\top}u\left(t\right)\end{aligned} \tag{54}$$

and the source term action is

$$\begin{aligned}Q\left[\ell,v,u\right]=&i\int_{0}^{\infty}dt\int_{0}^{t}dt'\left(K_{d}^{L}\left(t,t'\right)\right)^{\top}v\left(t'\right)\ell\left(t\right)\\&+i\int_{0}^{\infty}dt\int_{0}^{\infty}dt'm\left(t,t'\right)\left(k^{L}\left(t,t'\right)\right)^{\top}u\left(t'\right)\ell\left(t\right)\\&-\frac{1}{2}\int_{0}^{\infty}dt\int_{0}^{\infty}dt'm\left(t,t'\right)k^{L}\left(t,t',\mathbf{x},\mathbf{x}\right)\ell\left(t\right)\ell\left(t'\right)\end{aligned} \tag{55}$$

Where in the infinite width limit, we can identify $v(t)$ with $f_{\text{traim}}(t)$ by $iv_{t}=f_{\text{train}}\left(t\right)-Y$, which holds for all moments of $f_{\text{train}}(t)$.

For convenience, in the continuous time limit, we denote the NDK with a lower index $d$. The NDK in Eq.40 can be rewritten as

$$\mathcal{K}_{d}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=m\left(t,t'\right)\Delta^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)+e^{-T\sigma^{-2}|t-t'|}\mathcal{K}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right) \tag{56}$$

with

$$\begin{aligned}\Delta^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right) &= \frac{\lambda}{2T}\left(\mathcal{K}^{L,1}\left(t,t',\mathbf{x},\mathbf{x}'\right)-\mathcal{K}^{L,0}\left(t,t',\mathbf{x},\mathbf{x}'\right)\right) \\ &= \mathcal{K}^{d,L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\dot{\mathcal{K}}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)\end{aligned} \tag{57}$$

$$m\left(t,t'\right) = \sigma^{2}e^{-T\sigma^{-2}|t-t'|}+\left(\sigma_{0}^{2}-\sigma^{2}\right)e^{-T\sigma^{-2}(t+t')} \tag{58}$$

Here the quantity $m\left(t,t'\right)$ is the continuous time limit of $m^{1}_{t,t'}$. As defined in Eq.32, it represents the covariance of the prior

$$\left\langle \Theta^{i}_{t}\Theta^{j}_{t'}\right\rangle_{0} = \delta_{ij}m\left(t,t'\right), \left\langle \Theta^{i}_{t}\right\rangle_{0} = 0 \tag{59}$$

.

The above calculation leads to the recursion relation:

$$\begin{aligned}\mathcal{K}^{L}_{d}\left(t,t',\mathbf{x},\mathbf{x}'\right) =& m\left(t,t'\right)\mathcal{K}^{L-1}_{d}\left(t,t',\mathbf{x},\mathbf{x}'\right)\dot{\mathcal{K}}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right) \\ &+ e^{-T\sigma^{-2}|t-t'|}\mathcal{K}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)\end{aligned} \tag{60}$$

with initial condition

$$\mathcal{K}^{L=0}_{d}\left(t,t',\mathbf{x},\mathbf{x}'\right) = e^{-T\sigma^{-2}|t-t'|}\mathcal{K}_{in}\left(\mathbf{x},\mathbf{x}'\right) \tag{61}$$

Where $\mathcal{K}_{in}\left(\mathbf{x},\mathbf{x}'\right)$ was defined in Eq.39. We refer to this continuous time $K^{L}_{d}\left(t,t',\mathbf{x},\mathbf{x}'\right)$ as the Neural Dynamical Kernel (NDK). Note that it follows directly from Eq.60 that

$$\mathcal{K}^{L}_{d}\left(0,0,\mathbf{x},\mathbf{x}'\right) = \mathcal{K}^{L}_{NTK}\left(\mathbf{x},\mathbf{x}'\right). \tag{62}$$

For the mean predictor we use the results from the previous section Eqs.49,50,51, take the large $\lambda$ limit and turn the sums into integrals, we obtain

$$\left\langle f_{\text{train}}\left(t\right)\right\rangle = \int_{0}^{t} dt' K^{L}_{d}\left(t,t'\right)\left(Y-\left\langle f_{\text{train}}\left(t'\right)\right\rangle\right) \tag{63}$$

$$\left\langle f\left(t,\mathbf{x}\right)\right\rangle = \int_{0}^{t} dt'\left(k^{L}_{d}\left(t,t',\mathbf{x}\right)\right)^{\top}\left(Y-\left\langle f_{\text{train}}\left(t'\right)\right\rangle\right) \tag{64}$$

as given in Eqs.7, 8 in the main text.

## Appendix D. Second Moment

Our formalism allows for the derivation of higher moments of the predictor. In particular, we are interested in the covariance $\left\langle \delta f\left(t,\mathbf{x}\right)\delta f\left(t',\mathbf{x}'\right)\right\rangle \equiv \left\langle f\left(t,\mathbf{x}\right)f\left(t',\mathbf{x}'\right)\right\rangle - \left\langle f\left(t,\mathbf{x}\right)\right\rangle\left\langle f\left(t',\mathbf{x}'\right)\right\rangle$. We focus on the continuous time $\lambda\to\infty$ limit described in Sec.C.4, which is equivalent to Langevin dynamics. In order to calculate the second moment, we need to invert one time-dependent operator, which we denote as $B(t,t')\in\mathbb{R}^{P\times P}$:

$$B\left(t,t'\right) = I\delta\left(t-t'\right)+K^{L}_{d}\left(t,t'\right), \tag{65}$$

$$\int_0^t d\tau B\left(t,\tau\right) B^{-1}\left(\tau,t'\right) = I\delta\left(t-t'\right) \tag{66}$$

The full statistics of the Gaussian field $v(t), u(t)$ can be written in terms of $B^{-1}(t,t')$

$$\left\langle v\left(t\right)\right\rangle = i\int_0^t dt' B^{-1}\left(t,t'\right) Y \tag{67}$$

$$\left\langle \delta v\left(t\right)\delta v^{\top}\left(t'\right)\right\rangle = -\int_0^{\infty} d\tau' \int_0^{\infty} d\tau B^{-1}\left(t,\tau\right) m\left(\tau,\tau'\right) K^{L}\left(\tau,\tau'\right) B^{-1}\left(t',\tau'\right) \tag{68}$$

$$\left\langle v\left(t\right) u^{\top}\left(t'\right)\right\rangle = B^{-1}\left(t,t'\right) \tag{69}$$

It is useful to separate the smooth part from the delta function in the inverse operator $B^{-1}(t,t')$. We denote the smooth function as $J(t,t') \in \mathbb{R}^{P\times P}$, which satisfies the following integral equation:

$$J\left(t,t'\right) = \begin{cases} K_d^L\left(t,t'\right) - \int_{t'}^t d\tau K_d^L\left(t,\tau\right) J\left(\tau,t'\right) & t \geq t' \\ 0 & t < t' \end{cases} \tag{70}$$

$$B^{-1}\left(t,t'\right) = I\delta\left(t-t'\right) - J\left(t,t'\right) \tag{71}$$

We take the second derivative of the MGF (Eq.53):

$$\left\langle \delta f\left(\mathbf{x},t\right)\delta f\left(\mathbf{x}',t'\right)\right\rangle = \left.\frac{\partial^2 \mathcal{M}\left[\ell\right]}{\partial\ell\left(t,\mathbf{x}\right)\partial\ell\left(t',\mathbf{x}'\right)}\right|_{\ell(t,\mathbf{x})=\ell(t',\mathbf{x}')=0} - \left\langle f\left(\mathbf{x},t\right)\right\rangle\left\langle f\left(\mathbf{x}',t'\right)\right\rangle \tag{72}$$

Which we can express in terms of $J(t,t')$ using the derived statistics of $v(t)$,$u(t)$

$$\left\langle \delta f_{\text{train}}\left(t\right)\delta f_{\text{train}}^{\top}\left(t'\right)\right\rangle = m\left(t,t'\right) K^{L}\left(t,t'\right) - \int_0^t d\tau\left[J\left(t,\tau\right) m\left(t',\tau\right) K^{L}\left(t',\tau\right)\right] \tag{73}$$

$$-\int_0^{t'} d\tau\left[J\left(t',\tau\right) m\left(t,\tau\right) K^{L}\left(t,\tau\right)\right] + \int_0^t d\tau \int_0^{t'} d\tau'\left[J\left(t,\tau\right) m\left(\tau,\tau'\right) K^{L}\left(\tau,\tau'\right) J\left(t',\tau'\right)\right]$$

$$\left\langle \delta f\left(t,\mathbf{x}\right)\delta f\left(t',\mathbf{x}'\right)\right\rangle =\int_{0}^{t}d\tau\int_{0}^{t'}d\tau'[k_{d}^{L}\left(t,\tau,\mathbf{x}\right)^{\top}\left\langle \delta f_{train}\left(\tau\right)\delta f_{train}^{\top}\left(\tau'\right)\right\rangle k_{d}^{L}\left(t',\tau',\mathbf{x}'\right)] \quad (74)$$

$$+\int_{0}^{t}d\tau\int_{0}^{\tau}d\tau'[k_{d}^{L}\left(t,\tau,\mathbf{x}\right)^{\top}J\left(\tau,\tau'\right)m\left(t',\tau'\right)k^{L}\left(t',\tau',\mathbf{x}'\right)]$$

$$+\int_{0}^{t'}d\tau\int_{0}^{\tau}d\tau'[m\left(t,\tau\right)k^{L}\left(t,\tau,\mathbf{x}\right)^{\top}J\left(\tau,\tau'\right)k_{d}^{L}\left(t',\tau',\mathbf{x}'\right)]$$

$$-\int_{0}^{t}d\tau[k_{d}^{L}\left(t,\tau,\mathbf{x}\right)^{\top}m\left(t',\tau\right)k^{L}\left(t',\tau,\mathbf{x}'\right)]$$

$$-\int_{0}^{t'}d\tau[m\left(t,\tau\right)k^{L}\left(t,\tau,\mathbf{x}\right)^{\top}k_{d}^{L}\left(t',\tau,\mathbf{x}'\right)]+m\left(t,t'\right)\mathcal{K}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)$$

The equation becomes simpler for the correlation with initial condition, achieved by plugging $t'=0$ in Eq.74

$$\left\langle \delta f\left(t,\mathbf{x}\right)\delta f\left(t'=0,\mathbf{x}'\right)\right\rangle =m\left(t,0\right)\mathcal{K}^{L}\left(t,0,\mathbf{x},\mathbf{x}'\right)-\int_{0}^{t}d\tau[k_{d}^{L}\left(t,\tau,\mathbf{x}\right)^{\top}m\left(\tau,0\right)k^{L}\left(\tau,0,\mathbf{x}'\right)] \quad (75)$$

$$+\int_{0}^{t}d\tau\int_{0}^{\tau}d\tau'[k_{d}^{L}\left(t,\tau,\mathbf{x}\right)^{\top}J\left(\tau,\tau'\right)m\left(\tau',0\right)k^{L}\left(\tau',0,\mathbf{x}'\right)]$$

We note that the mean predictor can also be written using the $J(t,t')$ operator:

$$\left\langle f_{\text{train}}\left(t\right)\right\rangle =\int_{0}^{t}dt'J\left(t,t'\right)Y \quad (76)$$

$$\left\langle f\left(t,\mathbf{x}\right)\right\rangle =\int_{0}^{t}dt'\left[k_{d}^{L}\left(t,t',\mathbf{x}\right)^{\top}\left(I-\int_{0}^{t'}dt''J\left(t',t''\right)\right)\right]Y \quad (77)$$

Solving the integral equation for $J(t,t')$ for a general nonlinearity is complex. However, the equations are tractable in two cases: Linear networks and the NTK limit ($T\to 0, t\sim\mathcal{O}(1)$), which are presented below.

### D.1. The NTK Limit

The time dependence of all kernels arises from $m(t,t')$, and thus at the NTK limit, defined by $T\to 0, t\sim\mathcal{O}(1)$, we can substitute all the kernels and temporal correlations with their values at

initialization, specifically $K_d^L(t,t') \approx K_{NTK}^L, K(t,t') = K_{GP_0}, m(t,t') = \sigma_0^2$. Solving $J(t,t')$ with a constant NDK yields

$$J\left(t,t'\right) = \begin{cases} K_{NTK} \exp\left(-K_{NTK}^L\left(t-t'\right)\right) & t \geq t' \\ 0 & t < t' \end{cases} \tag{78}$$

The only time dependence in the covariance equation (Eq.74) comes from $J(t,t')$, as the kernels and $m(t,t')$ are constant. Performing the integral over the exponential $J(t,t')$ results in

$$\begin{aligned} &\lim_{T\to 0} \sigma_0^{-2} \left\langle \delta f\left(t,\mathbf{x}\right) \delta f\left(t',\mathbf{x}'\right)\right\rangle = \mathcal{K}_{GP_0}^L\left(\mathbf{x},\mathbf{x}'\right) - k_{GP_0}^L\left(\mathbf{x}\right)\left(K_{GP_0}^L\right)^{-1} k_{GP_0}^L\left(\mathbf{x}'\right) \\ &+ \left[\left(I - \exp\left(-K_{NTK}^L t\right)\right)\left(K_{NTK}^L\right)^{-1} k_{NTK}^L\left(\mathbf{x}\right) - \left(K_{GP_0}^L\right)^{-1} k_{GP_0}^L\left(\mathbf{x}\right)\right]^\top K_{GP_0}^L \\ &\cdot \left[\left(I - \exp\left(-K_{NTK}^L t'\right)\right)\left(K_{NTK}^L\right)^{-1} k_{NTK}^L\left(\mathbf{x}'\right) - \left(K_{GP_0}^L\right)^{-1} k_{GP_0}^L\left(\mathbf{x}'\right)\right] \end{aligned} \tag{79}$$

### D.2. Linear Network

For a linear network, the NDK can be written in terms of the sum of exponents (see Sec.E), and the integral equations for the first and second moments are tractable. We can represent both of them in terms of the function $J(t,t')$ (Eq.70)

$$\begin{aligned} &J\left(t,t'\right) = K_d^L\left(t',t'\right) \\ &\exp\left(-\left(L+1\right)\left(\left(K_{GP}^L + IT\sigma^{-2}\right)\left(t-t'\right) + \frac{1}{2T\sigma^{-2}} K_{GP}^L \sum_{n=1}^{L} \frac{L!}{n!\left(L-n\right)!} \frac{\gamma^n}{n}\left(e^{-2T\sigma^{-2}nt'} - e^{-2T\sigma^{-2}nt}\right)\right)\right) \end{aligned} \tag{80}$$

Where $K_{GP}^L = \sigma^{2L} K_{in}$, $K_{in}$ is defined in Eq.39 and $K_d^L(t,t')$ is given in linear network in Eq.86.

The mean predictor and the covariance can be calculated by substituting the expression for $J(t,t')$ into Eqs.74, 77, leading to integrals that can be evaluated numerically, rather than integral equations like in the nonlinear case.

**Low $T$ Limit:**

We can further simplify the expressions by taking the limit of $T \to 0$. In this limit, $J(t,t')$ is singular around $t = t'$ and is given by

$$J\left(t,t'\right) = T\sigma^{-2}\left(I\delta\left(t-t'\right) + T\sigma^{-2}\left(K_d^L\left(t,t\right)\right)^{-1}\left(\delta'\left(t-t'\right) + \left(L+1\right)\delta\left(t-t'\right)\right)\right) \tag{81}$$

Where $\delta(t-t')$ and $\delta'(t-t')$ are the Dirac delta function and its derivative, respectively. The leading order in $T$ of the mean predictor is

$$f\left(t,\mathbf{x}\right) = k_{in}\left(\mathbf{x}\right)^\top K_{in}^{-1}\left(I - \exp\left(-\left(L+1\right)\sigma_0^{2L} K_{in} t\right)\right) Y \tag{82}$$

It is important to note that in a linear network, the NTK equilibrium identifies with the NNGP equilibrium, and thus, the mean predictor dynamics are identical to the NTK dynamics, and reaches equilibrium at $t \sim \mathcal{O}(1)$.

The covariance equation in the low $T$ limit take the following simple form

$$\left\langle \delta f\left(t,\mathbf{x}\right) \delta f\left(t',\mathbf{x}'\right)\right\rangle = m^{L+1}\left(t,t'\right)\left[\mathcal{K}_{in}\left(\mathbf{x},\mathbf{x}'\right) - k_{in}\left(\mathbf{x}\right)^\top \left(K_{in}\right)^{-1} k_{in}\left(\mathbf{x}'\right)\right]$$

## Appendix E. The Neural Dynamical Kernel

We focus on the continuous time limit derived above, and present several examples where the NDK has explicit expressions, and provide proofs of properties of the NDK presented in the main text. We have derived

$$\begin{aligned}\mathcal{K}_d^L\left(t,t',\mathbf{x},\mathbf{x}'\right) =& m\left(t,t'\right)\mathcal{K}_d^{L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\dot{\mathcal{K}}^L\left(t,t',\mathbf{x},\mathbf{x}'\right)\\ &+e^{-T\sigma^{-2}|t-t'|}\mathcal{K}^L\left(t,t',\mathbf{x},\mathbf{x}'\right)\end{aligned} \tag{83}$$

In order to complete the calculation of the NDK, we would provide explicit analytical expressions for $\mathcal{K}(t,t',\mathbf{x},\mathbf{x}')$ and $\dot{\mathcal{K}}(t,t',\mathbf{x},\mathbf{x}')$ in cases where they are available, namely linear activation, and ReLU and error function nonlinearities.

### E.1. Linear Activation:

For linear activation:

$$\mathcal{K}^L\left(t,t',\mathbf{x},\mathbf{x}'\right) = \left(m\left(t,t'\right)\right)^L\mathcal{K}_{in}\left(\mathbf{x},\mathbf{x}'\right) \tag{84}$$

$$\dot{\mathcal{K}}^L\left(t,t',\mathbf{x},\mathbf{x}'\right) = I \tag{85}$$

The recursion relation for the NDK can be solved explicitly, yielding

$$\mathcal{K}_d^L\left(t,t',\mathbf{x},\mathbf{x}'\right) = \left(m\left(t,t'\right)\right)^L\left(L+1\right)e^{-T\sigma^{-2}|t-t'|}\mathcal{K}_{in}\left(\mathbf{x},\mathbf{x}'\right) \tag{86}$$

The NDK of linear activation is proportional to the input kernel $\mathcal{K}_{in}\left(\mathbf{x},\mathbf{x}'\right)$ regardless of the data. The effect of network depth only changes the magnitude but not the shape of the NDK. As a result, the NNGP and NTK kernels also only differ by their magnitude, and thus the mean predictor at the NNGP and NTK equilibria only differ by $\mathcal{O}\left(T\right)$. This suggests that the diffusive phase has very little effect on the mean predictor in the low $T$ regime, in linear network, as discussed in Sec.D.2.

### E.2. ReLU Activation:

For ReLU activation, we define the function $J\left(\theta\right)$ [7]:

$$J\left(\theta^L\left(t,t',\mathbf{x},\mathbf{x}'\right)\right) = \left(\pi-\theta^L\left(t,t',\mathbf{x},\mathbf{x}'\right)\right)\cos\left(\theta^L\left(t,t',\mathbf{x},\mathbf{x}'\right)\right)+\sin\left(\theta^L\left(t,t',\mathbf{x},\mathbf{x}'\right)\right) \tag{87}$$

where the angle between $\mathbf{x}$ and $\mathbf{x}'$ is given by :

$$\theta^L\left(t,t',\mathbf{x},\mathbf{x}'\right) = \cos^{-1}\left(\frac{m\left(t,t'\right)}{\sqrt{m\left(t,t\right)m\left(t',t'\right)}}\frac{1}{\pi}J\left(\theta^{L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\right)\right) \tag{88}$$

$\theta^L\left(t,t',\mathbf{x},\mathbf{x}'\right)$ is defined through a recursion equation, and

$$\theta^{L=0}\left(t,t',\mathbf{x},\mathbf{x}'\right) = \cos^{-1}\left(\frac{m\left(t,t'\right)}{\sqrt{m\left(t,t\right)m\left(t',t'\right)}}\frac{\mathcal{K}_{in}\left(\mathbf{x},\mathbf{x}'\right)}{\sqrt{\mathcal{K}_{in}(\mathbf{x},\mathbf{x})\mathcal{K}_{in}(\mathbf{x}',\mathbf{x}')}}\right) \tag{89}$$

the kernel functions are then given by

$$\dot{\mathcal{K}}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=\frac{1}{2\pi}\left(\pi-\theta^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)\right) \tag{90}$$

$$\mathcal{K}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=\frac{\sqrt{\mathcal{K}_{in}\left(\mathbf{x},\mathbf{x}\right)\mathcal{K}_{in}\left(\mathbf{x}',\mathbf{x}'\right)}}{\pi 2^{L}}\left(m\left(t,t\right)m\left(t',t'\right)\right)^{L/2}J\left(\theta^{L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\right) \tag{91}$$

We obtain an explicit expression for the NDK by plugging these kernels into Eqs.60,61.

### E.3. Error Function Activation

For error function activation [34]:

$$\mathcal{K}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=\frac{2}{\pi}\sin^{-1}\left(\frac{2m\left(t,t'\right)\mathcal{K}^{L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)}{\sqrt{\left(1+2m\left(t,t\right)\mathcal{K}^{L-1}\left(t,t,\mathbf{x},\mathbf{x}\right)\right)\left(1+2m\left(t',t'\right)\mathcal{K}^{L-1}\left(t',t',\mathbf{x}',\mathbf{x}'\right)\right)}}\right) \tag{92}$$

$$\begin{aligned}\dot{\mathcal{K}}^{L}_{\mu\nu}\left(t,t',\mathbf{x},\mathbf{x}'\right)=&\frac{4}{\pi}\left(\left(1+2m\left(t,t\right)\mathcal{K}^{L-1}\left(t,t,\mathbf{x},\mathbf{x}\right)\right)\left(1+2m\left(t',t'\right)\mathcal{K}^{L-1}\left(t',t',\mathbf{x}',\mathbf{x}'\right)\right)\right.\\&\left.-4\left(m\left(t,t'\right)\mathcal{K}^{L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\right)^{2}\right)^{-1/2}\end{aligned} \tag{93}$$

Again we can obtain an explicit expression for the NDK by plugging these kernels into Eqs.60,61.

### E.4. NDK as a Generalized Time-Dependent NTK

In Eq.9 in the main text, we claimed that the NDK has the following interpretation as a generalized two-time NTK

$$\mathcal{K}^{L}_{d}\left(t,t',\mathbf{x},\mathbf{x}'\right)=e^{-T\sigma^{-2}|t-t'|}\left\langle\nabla_{\Theta_t}f\left(\mathbf{x},\Theta_t\right)\cdot\nabla_{\Theta_{t'}}f\left(\mathbf{x}',\Theta_{t'}\right)\right\rangle_{0}\,t\geq t' \tag{94}$$

where $\langle\cdot\rangle_0$ denotes averaging w.r.t. the prior distribution of the parameters $\Theta$, with the statistics defined in Eq.59.

Now we provide a formal proof.

We separate $\nabla_{\Theta_t}f\left(\mathbf{x},\Theta_t\right)$ into two parts including the derivative w.r.t. the readout weights $a_t$ and the hidden layer weights $\mathbf{W_t}$

**Derivative w.r.t. the readout weights:**

$$\left\langle\partial_{\mathbf{a}_t}f\left(\mathbf{x},\Theta_t\right)\cdot\partial_{\mathbf{a}_{t'}}f\left(\mathbf{x},\Theta_{t'}\right)\right\rangle_{0}=\mathcal{K}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right) \tag{95}$$

**Derivative w.r.t. the hidden layer weights:**

We have

$$\partial_{\mathbf{W}^{l}_{t}}\mathbf{x}^{L}_{t}\left(\mathbf{x},\mathbf{W}_t\right)=\frac{1}{\sqrt{N_{L-1}\cdots N_{l-1}}}\Pi^{L}_{k=l+1}\left[\phi'\left(z^{k}_{t}\right)\mathbf{W}^{k}_{t}\right]\phi'\left(z^{l}_{t}\right)\mathbf{x}^{l-1}_{t} \tag{96}$$

and

$$\begin{aligned}&\left\langle\partial_{\mathbf{W}^{l}_{t}}f\left(\mathbf{x},\Theta_t\right)\cdot\partial_{\mathbf{W}^{l}_{t'}}f\left(\mathbf{x},\Theta_{t'}\right)\right\rangle_{0}\\&=\left\langle N_{L}^{-1}\mathbf{a}_t\cdot\mathbf{a}_{t'}\right\rangle\left(\Pi^{L}_{k=l+1}\left\langle N_{k}^{-1}N_{k-1}^{-1}\mathbf{W}^{k}_{t}\cdot\mathbf{W}^{k}_{t'}\right\rangle\right)\left(\Pi^{L}_{k=l}\dot{\mathcal{K}}^{k}\left(t,t',\mathbf{x},\mathbf{x}'\right)\right)\mathcal{K}^{l-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\\&=m\left(t,t'\right)^{L-l+1}\left(\Pi^{L}_{k=l}\dot{\mathcal{K}}^{k}\left(t,t',\mathbf{x},\mathbf{x}'\right)\right)\mathcal{K}^{l-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\end{aligned} \tag{97}$$

To leading order in $N_l$ the averages over $\mathbf{a}$ and $\mathbf{W}$ can be performed separately for each layer, and are dominated by their prior, where each element of the weights is an independent Gaussian given by Eq.27. The term $m\left(t,t'\right)$ comes from the covariance of the priors in $\mathbf{W}$ and $\mathbf{a}$, since there are a total of $L-l$ layers of $\mathbf{W}$ and one layer of $\mathbf{a}$, we have $m\left(t,t'\right)^{L-l+1}$. The kernel $\dot{\mathcal{K}}^{k}\left(t,t',\mathbf{x},\mathbf{x}'\right)$ comes from the inner product between $\phi'\left(z_{t}^{k}\right)$ and $\phi'\left(z_{t'}^{k}\right)$, and the kernel $\mathcal{K}^{l-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)$ comes from the inner product between $\mathbf{x}_{t}^{l-1}$ and $\mathbf{x}_{t'}^{l-1}$.

Using proof by induction as for the NTK [15], we obtain

$$\left\langle \partial_{\mathbf{W}_{t}}f\left(\mathbf{x},\Theta_{t}\right)\cdot\partial_{\mathbf{W}_{t'}}f\left(\mathbf{x},\Theta_{t'}\right)\right\rangle_{0}=e^{T\sigma^{-2}|t-t'|}m\left(t,t'\right)\dot{\mathcal{K}}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)\mathcal{K}^{d,L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right) \tag{98}$$

Combine Eq.98 with Eq.95 and with the definition of $\mathcal{K}_{d}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)$ in Eq.60, we have

$$e^{-T\sigma^{-2}|t-t'|}\left\langle \nabla_{\Theta_{t}}f\left(\mathbf{x},\Theta_{t}\right)\cdot\nabla_{\Theta_{t'}}f\left(\mathbf{x}',\Theta_{t'}\right)\right\rangle_{0}=\mathcal{K}_{d}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right) \tag{99}$$