CONNECTING NTK AND NNGP: A UNIFIED THEORETICAL FRAMEWORK FOR NEURAL NETWORK LEARNING DYNAMICS IN THE KERNEL REGIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Artificial neural networks (ANNs) have revolutionized machine learning in recent years, but a complete theoretical framework for their learning process is still lacking. Substantial theoretical advances have been achieved for infinitely wide networks. In this regime, two disparate theoretical frameworks have been used, in which the network's output is described using kernels: one framework is based on the Neural Tangent Kernel (NTK), which assumes linearized gradient descent dynamics, while the Neural Network Gaussian Process (NNGP) kernel assumes a Bayesian framework. However, the relation between these two frameworks and between their underlying sets of assumptions has remained elusive. This work unifies these two distinct theories using gradient descent learning dynamics with an additional small noise in an ensemble of randomly initialized infinitely wide deep networks. We derive an exact analytical expression for the network input-output function during and after learning and introduce a new time-dependent Neural Dynamical Kernel (NDK) from which both NTK and NNGP kernels can be derived. We identify two important learning phases characterized by different time scales: gradient-driven and diffusive learning. In the initial gradient-driven learning phase, the dynamics is dominated by deterministic gradient descent, and is adequately described by the NTK theory. This phase is followed by the slow diffusive learning stage, during which the network parameters sample the solution space, ultimately approaching the equilibrium posterior distribution corresponding to NNGP. Combined with numerical evaluations on synthetic and benchmark datasets, we provide novel insights into the different roles of initialization, regularization, and network depth, as well as phenomena such as early stopping and representational drift. This work closes the gap between the NTK and NNGP theories, providing a comprehensive framework for understanding the learning process of deep neural networks in the infinite width limit.

1 INTRODUCTION

Despite the empirical success of artificial neural networks (ANNs), theoretical understanding of their underlying learning process is still limited. One promising theoretical approach focuses on deep wide networks, in which the number of parameters in each layer goes to infinity while the number of training examples remains finite (Jacot et al. (2018); Lee et al. (2018; 2019); Novak et al. (2018; 2019); Matthews et al. (2018); Yang (2019); Sohl-Dickstein et al. (2020)). In this regime, the neural network (NN) is highly over-parameterized, and there is a degenerate space of solutions achieving zero training error. Investigating the properties of the solution space offers an opportunity for understanding learning in over-parametrized NNs (Chizat & Bach (2020); Jin & Montúfar (2020); Min et al. (2021)). The two well-studied theoretical frameworks in the infinite width limit focus on two different scenarios for exploring the solution space during learning. One considers randomly initialized NNs trained with gradient descent dynamics, and the learned NN parameters are largely dependent on their value at initialization. In this case, the infinitely wide NN's input-output relation is captured by the neural tangent kernel (NTK) (Jacot et al. (2018); Lee et al. (2019)). The other scenario considers Bayesian neural networks (BNNs) with an i.i.d. Gaussian prior over their parameters, and a learning-induced posterior distribution. In this case, the statistics of the NN's input-output relation in the infinite width limit are given by the neural network Gaussian

process (NNGP) kernel (Cho & Saul (2009); Lee et al. (2018)). These two scenarios make different assumptions regarding the learning process and regularization. Furthermore, for some datasets the generalization performance of the two kernels differs significantly (Lee et al. (2020)). It is therefore important to generate a unified framework with a single set of priors and regularizations describing a dynamical process that captures both cases. Such a theory may also provide insight into salient dynamical phenomena such as early stopping (Li et al. (2020); Advani et al. (2020); Ji et al. (2021)). From a neuroscience perspective, a better understanding of the exploratory process leading to Bayesian equilibrium may shed light on the empirical and hotly debated phenomenon of representational drift (Rokni et al. (2007); Rule et al. (2019); Deitch et al. (2021); Marks & Goard (2021); Schoonover et al. (2021)). To this end, we construct a new analytical theory of the learning dynamics in infinitely wide ANNs. Our main contributions are:

1. We derive an analytical expression for the time evolution of the mean input-output relation (i.e. the mean predictor) of infinitely wide networks under Langevin dynamics in the form of an integral equation, and demonstrate its remarkable agreement with computer simulations.

2. A new two-time kernel, the Neural Dynamical Kernel (NDK), naturally emerges from our theory and we derive explicit relations between the NDK and both the NTK and the NNGP kernels.

3. Our theory reveals two important learning phases characterized by different time scales: gradient-driven and diffusive learning. In the initial gradient-driven learning phase, the dynamics is primarily governed by deterministic gradient descent, and can be described by the NTK theory. This phase is followed by a slow diffusive stage, during which the network parameters sample the solution space, ultimately approaching the equilibrium posterior distribution corresponding to NNGP.

4. We apply our theory to both synthetic and benchmark datasets and present several predictions. Firstly, the generalization error may exhibit diverse behaviors during the diffusive learning phase depending on network depth and the ratio between initialization and regularization strengths. Our theory provides insights into the roles of these hyper-parameters in early stopping. Secondly, through analysis of the temporal correlation between network weights during diffusive learning, we show that despite the random diffusion of hidden layer weights, the training error remains stable at a very low value due to a continuous realignment of readout weights and network hidden layer weights. Conversely, a time delay in this alignment degrades the network performance due to decorrelation in the representation, ultimately leading to degraded performance. We derive conditions under which the performance upon completely decorrelated readout and hidden weights remain well above chance. This provides insight into the representational drift and its consequences observed in biological neuronal circuits.

5. Relation to previous work: Previous work considered a single-time NTK kernel Jacot et al. (2018). This implicit time dependence arises through the weight dependence of an uaveraged weight-dependent NTK. Our (two-) time dependence arises in an appropriate weight-averaged kernel, and therefore exhibits explicit time dependence. Other perspective on the two phases has been proposed by previous works, Shwartz-Ziv & Tishby (2017) establishes an information theory framework to describe the two phases, Li et al. (2021); Blanc et al. (2020) analyze the two phases in SGD, where the diffusive phase is driven by the different types of noise (label noise or isotropic noise). Our work complements these previous findings by providing theoretical analysis of the two phases under Langevin dynamics, reaffirming the connections between the diffusive learning stage and representational drift in neuroscience as established in previous works Aitken et al. (2022); Pashakhanloo & Koulakov (2023).

2 Theoretical Results

In this section, we present our dynamic theory for infinitely wide deep networks under Langevin dynamics. We define a new time-dependent kernel, the Neural Dynamical Kernel (NDK), and derive an exact analytical integral equation for the mean predictor of the network.

2.1 Notation and setup for architecture and training dynamics

We consider a fully connected DNN with L hidden layers with a vector input $\mathbf{x} \in \mathbb{R}^{N_0}$ and a single output $\mathrm{f}(\mathbf{x}, t)$ (i.e., the predictor), with the following time-dependent input-output function:

$$\mathrm{f}(\mathbf{x}, t) = \frac{1}{\sqrt{N_L}} \mathbf{a}(t) \cdot \mathbf{x}^L(\mathbf{x}, t), \quad \mathbf{a}(t) \in \mathbb{R}^{N_L} \tag{1}$$

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

Here N_l denotes the number of nodes in hidden layer l, and N_0 is the input dimension. The set of network weights at a training time t is denoted collectively as $\boldsymbol{\theta}(t) = \left\{\mathbf{W}^1(t) \cdots \mathbf{W}^L(t), \mathbf{a}(t)\right\}$, where $\mathbf{a}(t) \in \mathbb{R}^{N_L}$ denotes the linear readout weights and $\mathbf{W}(t)^l \in \mathbb{R}^{N_l \times N_{l-1}}$ the hidden layer weights between layer $l-1$ and l. $\phi\left(N_{l-1}^{-1/2}\mathbf{W}^l(t) \cdot \mathbf{x}^{l-1}(\mathbf{x}, t)\right)$ is an element-wise nonlinear function of the weighted sum of its input vector, and $\mathbf{x}^{l=0} \equiv \mathbf{x}$ is the input to the first layer. The training data is a set of P labeled examples $\mathcal{D} : \{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\cdots,P}$ where there are P training input vectors $\mathbf{x}^\mu \in \mathbb{R}^{N_0}$, and $\boldsymbol{y} \in \mathbb{R}^P$ is a vector of the target labels of the training examples. We denote $\mathbf{f}_{\text{train}}(t) \in \mathbb{R}^P$, a vector containing the predictor on the P training vectors. We consider the supervised learning cost function:

$$E(\boldsymbol{\theta}(t)|\mathcal{D}) = \frac{1}{2}\left|\mathbf{f}_{\text{train}}(t) - \boldsymbol{y}\right|^2 + \frac{T}{2\sigma^2}\left|\boldsymbol{\theta}(t)\right|^2 \tag{3}$$

The first term is the squared error empirical loss (SE loss), and the second term is a regularization term that favors weights with small L_2 norm, where $|\boldsymbol{\theta}(t)|^2$ is the sum of the squares of all weights. It is convenient to introduce the temperature parameter (see definition below) T as controlling the relative strength of the regularization, and σ^2 is the variance of the equilibrium distribution of the Gaussian prior. We consider noisy gradient descent learning dynamics given by continuous-time Langevin dynamics, where the weights of the system start from an i.i.d. Gaussian initial condition with zero mean and variance σ_0^2, and evolve under gradient descent dynamics with respect to the cost function above with an additive white noise $\boldsymbol{\xi}(t)$:

$$\frac{d}{dt}\boldsymbol{\theta}(t) = -\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}(t)) + \boldsymbol{\xi}(t) \tag{4}$$

where $\boldsymbol{\xi}(t)$ has a white noise statistics $\mathbb{E}\left[\boldsymbol{\xi}(t)\boldsymbol{\xi}(t')^\top\right] = 2T\boldsymbol{I}\delta(t-t'), \mathbb{E}\left[\boldsymbol{\xi}(t)\right] = 0$, and T is the temperature controlling the level of noise in the system. As we show below, the additional white noise compared to deterministic gradient descent allows for continued exploration of the solution space after reaching a small training error, and enables the connection from NTK to NNGP theory.

2.2 Infinite Width Limit

We are interested in the predictor statistics (in particular the mean predictor) induced by the Langevin dynamics, which can be evaluated analytically in the infinite width where the hidden layer widths are taken to infinity, while the number of training examples P remains finite. For simplicity, we consider all the N_l to be the same for $l = 1, \cdots, L$ and equal to N, $N \to \infty$.

SI Sec.A presents a derviation of a path integral formulation of the above Langevin dynamics using a Markov proximal learning framework. Evaluating statistical quantities using these integrals is in general intractable. However, in the infinite width limit, they become tractable, as proven in SI Sec.B.1-B.4. Specifically, the moments of the predictor can be derived from a moment generating function (MGF) $\mathcal{M}[\ell(t)]$, written in the form of a path integral over two auxiliary time-dependent vectors, $\mathbf{u}(t) \in \mathbb{R}^P$ and $\mathbf{v}(t) \in \mathbb{R}^P$. Additionally, $\tilde{\mathbf{u}}(t) = [\mathbf{u}(t), i\ell(t)] \in \mathbb{R}^{P+1}$, where $\ell(t)$ denotes the field coupled to the predictor $f(\mathbf{x}, t)$ on an arbitrary test input vector $\mathbf{x}$. Therefore, the moments of the predictor can be derived by evaluating the derivative of $\mathcal{M}[\ell(t)]$ at $\ell(t) = 0$. In SI Sec.B.4 it is shown that the MGF has the following form,

$$\mathcal{M}[\ell(t)] = \int D\mathbf{u}(t) \int D\mathbf{v}(t) \exp\left(-S[\mathbf{v}(t), \tilde{\mathbf{u}}(t)]\right) \tag{5}$$

$$\begin{aligned} S[\mathbf{v}(t), \tilde{\mathbf{u}}(t)] = &\frac{1}{2}\int_0^\infty dt \int_0^\infty dt' m(t,t')\, \tilde{\mathbf{u}}^\top(t)\, \tilde{\mathbf{K}}^L(t,t')\, \tilde{\mathbf{u}}(t') \\ &+ \int_0^\infty dt \int_0^t dt' \tilde{\mathbf{u}}(t)^\top \tilde{\mathbf{K}}^{d,L}(t,t')\, \mathbf{v}(t') + \int_0^\infty dt \mathbf{u}(t)^\top (\mathbf{v}(t) - i\boldsymbol{y}) \end{aligned} \tag{6}$$

where $\int D\mathbf{u}(t)$ means integration over all trajectories of $\mathbf{u}$ and similarly for $\mathbf{v}$. The two-time kernel matrices $\tilde{\mathbf{K}}^L(t,t') \in \mathbb{R}^{(P+1)\times(P+1)}$ and $\tilde{\mathbf{K}}^{d,L}(t,t') \in \mathbb{R}^{(P+1)\times P}$ in Eq.6 are defined by applying the kernel functions $K^L(t,t',\mathbf{x},\mathbf{x}')$ and $K^{d,L}(t,t',\mathbf{x},\mathbf{x}')$ on P training data $\mathbf{x}_\mu, 1 \le \mu \le P$ and a single test point, $\mathbf{x}_{P+1} = \mathbf{x}$. Specifically, $\tilde{\mathbf{K}}^L_{\mu,\nu}(t,t') = \mathbf{K}^L(\mathbf{x}_\mu, \mathbf{x}_\nu, t, t'), 1 \le \mu, \nu \le P+1$ and

$\tilde{\mathbf{K}}^{d,L}_{\mu,\nu} = \tilde{\mathbf{K}}^{d,L}(\mathbf{x}_\mu, \mathbf{x}_\nu, t, t'), 1 \leq \mu \leq P+1, 1 \leq \nu \leq P$. $m(t,t')$ is a two-time dependent scalar function.

We first provide expressions of these kernel functions as well as the scalar two-time function $m(t,t')$ in Sec.2.3; we then present an expression for the mean predictor in terms of these kernels in Sec.2.4. The full derivation of these equations are given in Sec. B.1-B.4.

2.3 The Neural Dynamical Kernel (NDK)

The kernel function $K^{d,L}(t,t',\mathbf{x},\mathbf{x}')$ in Eq.6 is a new kernel function in our theory, denoted as the Neural Dynamical Kernel (NDK), which can be viewed as a time-dependent generalization of the NTK, and can be expressed in terms of derivatives of the predictor w.r.t. the time-dependent network parameters (SI Sec.C.4):

$$K^{d,L}(t,t',\mathbf{x},\mathbf{x}') = e^{-T\sigma^{-2}|t-t'|}\mathbb{E}_{\boldsymbol{\theta}\sim S_0}\left[\nabla_{\boldsymbol{\theta}(t)}\mathrm{f}(\mathbf{x},t)\cdot\nabla_{\boldsymbol{\theta}(t')}\mathrm{f}(\mathbf{x}',t')\right] \tag{7}$$

where S_0 denotes a Gaussian probability measure of the weights

$$\mathbb{E}_{\boldsymbol{\theta}\sim S_0}\left[\boldsymbol{\theta}(t)\,\boldsymbol{\theta}(t')^\top\right] = m(t,t')\,\boldsymbol{I}, \mathbb{E}_{\boldsymbol{\theta}\sim S_0}\left[\boldsymbol{\theta}(t)\right] = 0 \tag{8}$$

$$m(t,t') = \sigma^2 e^{-T\sigma^{-2}|t-t'|} + \left(\sigma_0^2 - \sigma^2\right)e^{-T\sigma^{-2}\left(t+t'\right)} \tag{9}$$

Here and in Eq.7 T is the level of noise in the Langevin dynamics, σ^2 and σ_0^2 are the variances of the L_2 regularizer and initial weights distribution, respectively. As expected, $m(0,0) = \sigma_0^2$ is the variance of the weights at initialization. At long times, the last (transient) term in Eq.9 vanishes and the first term dominates, such that $m(t,t')$ and $\mathrm{K}^{d,L}(t,t',\mathbf{x},\mathbf{x}')$ become functions of time difference $|t-t'|$. From Eqs.7, 9 follow that at initialization

$$K^{d,L}(0,0,\mathbf{x},\mathbf{x}') = \mathbb{E}_{\boldsymbol{\theta}_0\sim\mathcal{N}\left(0,\boldsymbol{I}\sigma_0^2\right)}\left[\nabla_{\boldsymbol{\theta}_0}\mathrm{f}(\mathbf{x},0)\cdot\nabla_{\boldsymbol{\theta}_0}\mathrm{f}(\mathbf{x}',0)\right] = K^L_{NTK}(\mathbf{x},\mathbf{x}') \tag{10}$$

The NDK equals the NTK as the average is only over the i.i.d. Gaussian initialization. Furthermore, as we will see in Sec.3.2, the NNGP kernel can also be obtained from the NDK. The other kernel function $K^L(t,t',\mathbf{x},\mathbf{x}')$ that appears in Eq.6 is a two-time extension of the NNGP kernel function

$$K^L(t,t',\mathbf{x},\mathbf{x}') = \mathbb{E}_{\boldsymbol{\theta}\sim S_0}\left[N^{-1}\mathbf{x}^L(\mathbf{x},t)\cdot\mathbf{x}^L(\mathbf{x}',t')\right] \tag{11}$$

The NDK defined in Eq.7 can be computed recursively, in terms of the two-time NNGP kernel $K^L(t,t',\mathbf{x},\mathbf{x}')$ and the derivative kernel $\dot{K}^L(t,t',\mathbf{x},\mathbf{x}')$ (see SI 2.3 for a detailed proof of the equivalence), given by

$$\begin{aligned}
&K^{d,L}(t,t',\mathbf{x},\mathbf{x}') = m(t,t')\,\dot{K}^L(t,t',\mathbf{x},\mathbf{x}')\,K^{d,L-1}(t,t',\mathbf{x},\mathbf{x}') + e^{-T\sigma^{-2}|t-t'|}K^L(t,t',\mathbf{x},\mathbf{x}')\\
&K^{d,L=0}(t,t',\mathbf{x},\mathbf{x}') = e^{-T\sigma^{-2}|t-t'|}\left(N_0^{-1}\mathbf{x}\cdot\mathbf{x}'\right)
\end{aligned} \tag{12}$$

The derivative kernel, $\dot{K}^L(t,t',\mathbf{x},\mathbf{x}')$ is the kernel evaluated w.r.t. the derivative of the activation functions

$$\dot{K}^L(t,t',\mathbf{x},\mathbf{x}') \equiv \mathbb{E}_{\boldsymbol{\theta}\sim S_0}\left[N^{-1}\dot{\mathbf{x}}^L(\mathbf{x},t)\cdot\dot{\mathbf{x}}^L(\mathbf{x}',t')\right] \tag{13}$$

where $\dot{\mathbf{x}}^L(\mathbf{x},t) = \phi'\left(N_{L-1}^{-\frac{1}{2}}\mathbf{W}_t^L\cdot\mathbf{x}^{L-1}(\mathbf{x},t)\right)$, namely $d\phi(z)/dz$ evaluated at the preactivtion of $\mathbf{x}^{L-1}$. All the kernel functions above including the two-time NNGP kernel, the derivative kernel, and the NDK, have closed-form expressions for specific activation functions such as linear, ReLU and error function (see SI Sec.C.1-C.3, Cho & Saul (2009); Williams (1996)).

2.4 Equations for the mean predictor

Here we provide the expressions of the mean predictor averaged over the distribution of learning trajectories, under the Langevin dynamics (Eq.4), using the NDK introduced above. The mean predictor can be derived by evaluating the derivative of the MGF (Eq.37, see details in SI Sec.B.3) The mean predictor on the training inputs obeys the following integral equation

$$\mathbb{E}\left[\mathbf{f}_{\text{train}}(t)\right] = \int_0^t dt'\mathbf{K}^{d,L}(t,t')\left(\boldsymbol{y} - \mathbb{E}\left[\mathbf{f}_{\text{train}}(t')\right]\right) \tag{14}$$

and the mean predictor on any test point $\mathbf{x}$ is given by an integral over the training predictor with the NDK of the test data $\mathbf{x}$

$$\mathbb{E}\left[\mathrm{f}\left(\mathbf{x},t\right)\right]=\int_{0}^{t}dt'\mathbf{k}^{d,L}\left(t,t'\right)^{\top}\left(\boldsymbol{y}-\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t'\right)\right]\right) \tag{15}$$

Here we have introduced separate notations for the kernel function applied on training data and testing data, $\mathbf{K}^{d,L}(t,t')\in\mathbb{R}^{P\times P}$ and $\mathbf{k}^{d,L}(t,t')\in\mathbb{R}^{P}$, defined as $\mathbf{K}_{\mu,\nu}^{d,L}(t,t')\equiv\tilde{\mathbf{K}}_{\mu,\nu}^{d,L}(t,t')$ and $\mathbf{k}_{\mu}^{d,L}(t,t')\equiv\tilde{\mathbf{K}}_{P+1,\mu}^{d,L}(t,t')$, respectively. Throughout the paper we will use similar notations for these kernel matrices and vectors (i.e., $\mathbf{K}\in\mathbb{R}^{P\times P}$ for kernel functions applied on training data, and $\mathbf{k}\in\mathbb{R}^{P}$ for kernel functions applied on test and training data).

3 Correspondence to NTK and NNGP

In this section, we show that our theory recovers known results of the NTK (Jacot et al. (2018)) and NNGP theories (Lee et al. (2018)) in the short- and long-time limit, respectively. We stress that the separation of time scales occurs in the limit of small but nonzero noise controlled by T, which is also the relevant limit of a realistic machine-learning scenario.

3.1 Gradient-driven phase corresponds to NTK dynamics

The time dependence of the NDK (Eq.12) is in time scales of $T\cdot t$ (Eqs.9,12), and thus at low T and $t\sim\mathcal{O}\left(1\right)$ we can substitute $K^{d,L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=K^{d,L}\left(0,0,\mathbf{x},\mathbf{x}'\right)$. In Sec.2.3, we obtain an exact equivalence between the NDK at time zero and the NTK. In this regime, $\mathbf{K}^{d,L}(t,t')$ and $\mathbf{k}^{d,L}(t,t')$ in Eq.15 and Eq.14 become time-independent. By taking the time derivative on both sides, the integral equations Eq.15 and Eq.14 can be transformed into a linear ODE, and solved analytically, leading to the well-known mean predictor in the NTK theory:

$$\mathbb{E}\left[\mathrm{f}\left(\mathbf{x},t\right)\right]\approx{\mathbf{k}_{NTK}^{L}}^{\top}\left[\mathbf{K}_{NTK}^{L}\right]^{-1}\left(I-\exp\left(-\mathbf{K}_{NTK}^{L}t\right)\right)\boldsymbol{y},t\sim\mathcal{O}\left(1\right) \tag{16}$$

where we define $\mathbf{k}_{NTK}^{L}\in\mathbb{R}^{P}$ and $\mathbf{K}_{NTK}^{L}\in\mathbb{R}^{P\times P}$ as the NTK applied on test and training data, respectively, similar to Sec.2.3. We see that the NTK theory describes the dynamics of the system when the time is short compared to the level of noise, such that the dynamics is approximately deterministic. Taking the large t limit of the NTK dynamics (Eq.16) results in the "NTK equilibrium", where $\lim_{t\to\infty}\mathbb{E}\left[\mathrm{f}\left(\mathbf{x},t\right)\right]={\mathbf{k}_{NTK}^{L}}^{\top}\left[\mathbf{K}_{NTK}^{L}\right]^{-1}\boldsymbol{y}$. This short-time equilibrium marks the crossover between the gradient-driven phase and the diffusive learning phase. After the NTK equilibrium point, the gradient of the loss is $\mathcal{O}\left(T\right)$, and thus the two parts of the cost function in Eq.3 (the SE loss and the regularization) are on equal footing, and give rise to the diffusive dynamics in time scales of $t\sim\mathcal{O}\left(T^{-1}\right)$.

3.2 Long-time equilibrium corresponds to NNGP

Now we investigate the behavior at long time scales defined by $t,t'\gg T^{-1}$. In this regime, $K^{d,L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=K^{d,L}\left(t-t',\mathbf{x},\mathbf{x}'\right)$ is a function of the time difference, and the transient dependence on the initialization parameter σ_0 vanishes. Furthermore, in this regime the limit of the integral of the NDK (Eq.7, Eq.12) satisfies the following identity (see SI Sec.C.4 for detailed proof):

$$\lim_{t\to\infty}\left(\int_{0}^{t}K^{d,L}\left(t-t',\mathbf{x},\mathbf{x}'\right)dt'\right)=\sigma^{2}T^{-1}K_{GP}^{L}\left(\mathbf{x},\mathbf{x}'\right) \tag{17}$$

where $K_{GP}^{L}\left(\mathbf{x},\mathbf{x}'\right)=\mathbb{E}_{\mathbf{W}\sim\mathcal{N}(0,\sigma^{2}I)}\left[N^{-1}\mathbf{x}^{L}\left(\mathbf{x}\right)\cdot\mathbf{x}^{L}\left(\mathbf{x}'\right)\right]$ is the well-known NNGP kernel. As a result, the mean predictor on arbitrary input $\mathbf{x}$ (Eq.15) correspondingly converges to an equilibrium (see SI C.4),

$$\lim_{t\to\infty}\mathbb{E}\left[\mathrm{f}\left(\mathbf{x},t\right)\right]={\mathbf{k}_{GP}^{L}}^{\top}\left(\boldsymbol{I}T\sigma^{-2}+\mathbf{K}_{GP}^{L}\right)^{-1}\boldsymbol{y} \tag{18}$$

where $\mathbf{k}_{GP}^{L}\in\mathbb{R}^{P}$ is the NNGP kernel function applied to $\mathbf{x}$. This is the known equilibrium NNGP result (Lee et al. (2018)). We emphasize that this result is true for any temperature, while the NTK solution in Sec.3.1 is relevant at low T only. Our theory thus establishes the connection between the NTK and the NNGP theories.

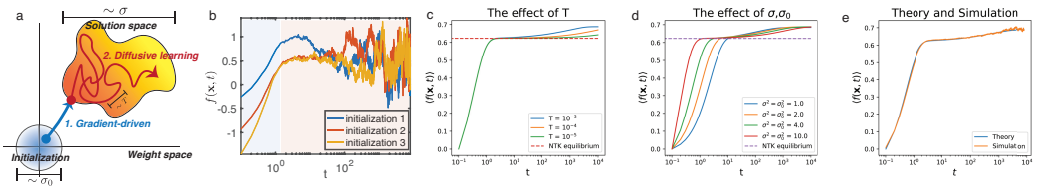

Figure 1: Two phases of the dynamics. Example using a synthetic dataset where the training inputs are orthogonal to each other with random binary labels $y^\mu \in \{\pm 1\}$. Each test point has partial overlap with one input point and is orthogonal to all the others. The desired test label is the same as the label on the training input with which it has nonzero overlap. (a) Schematics of the dynamics, the weights are initialized with width σ_0. The gradient-driven dynamics bring the weights to the solution space with a small training error, and the diffusive learning dynamics explores the solution space with a time scale T^{-1}. (b) Three example trajectories of $\mathrm{f}(\mathbf{x}, t)$, the dynamics are initially deterministic, and fluctuate significantly when t is large. (c-d) The network mean predictor on a test point with the desired label $+1$ (see details in SI Sec.E). (c) T does not affect the initial gradient-driven phase, but decreasing T slows the dynamics of the diffusive learning phase. (d) Increasing σ^2 and σ_0^2 simultaneously (keeping $\sigma^2 = \sigma_0^2$) affects the time scales of the two phases differently. The time scale of the gradient-driven phase decreases as σ_0^2 increases and vice versa in the diffusive dynamics. (e) The mean predictor under Langevin simulations of neural networks for the synthetic dataset agrees well with the theory prediction.

4 Dynamics at low T

In this section, we study the equations for the mean predictor dynamics (Eqs. 14, 15) in the important limit of low T. As we show below and illustrate in Fig.1(a,b), the network dynamics exhibits two distinct regimes. First, the network weights are initialized with width σ_0, and converge to weights with almost zero training error (error of $\mathcal{O}(T)$) approximately deterministically. Subsequently, the network executes slow and noise-driven explorations (on a time scale of $\mathcal{O}\left(T^{-1}\right)$) of the solution space, regularized by a Gaussian prior with width σ. We investigate how the different parameters such as initialization, regularization and the level of noise affect the learning behavior by evaluating numerically Eqs. 14, 15.

4.1 Time scales of the dynamics

In this section, we further examine how the time scales of the dynamics in the two phases are affected by the different hyper-parameters. We focus on the level of stochasticity T, the initialization (σ_0^2), and regularization (σ^2). As can be seen in Eqs.9, 12, the dynamics depend on t through exponents $\exp\left(-T\sigma^{-2}t\right)$ and a scalar factor that depends on σ_0^2/σ^2. To determine the time scales of the dynamics, we fix the scalar factor σ_0^2/σ^2 as a constant as we vary σ_0^2, σ^2 and T respectively. We consider $\sigma_0^2, \sigma^2 \sim \mathcal{O}(1)$.

First, we evaluate how the dynamics depends on the level of stochasticity determined by a small but nonzero T. As we see in Fig.1 (c), while the initial learning phase is not affected by T since the dynamics is mainly driven by deterministic gradient descent, the diffusive phase is slower for smaller T since it is driven by noise. We then investigate how the dynamics depends on σ^2 and σ_0^2 while fixing the ratio between them. Fig.1 (d) shows that as we increase σ^2 and σ_0^2 simultaneously, the gradient dynamics becomes faster since the initialization weights determined by σ_0^2 are closer to the typical solution space (with the L_2 regularization), while the dynamics of the diffusive phase becomes slower since the regularization determined by σ^2 imposes less constraint on the solution space, hence exploration time increases.

4.2 Diffusive learning dynamics exhibit diverse behaviors

In this section, we focus on the diffusive phase, where $t \sim \mathcal{O}(1/T)$. Unlike the simple exponential relaxation of the gradient-driven stage, in the diffusive phase, the predictor dynamics exhibits complex behavior dependent on depth, regularization, initialization and the data. We systematically explore these behaviors by solving the integral equations (Eqs.14, 15) numerically for benchmark datasets as well as a simplified synthetic task (see details of the tasks in Fig.1,2 captions and SI Sec.E). We verify the theoretical predictions with simulations of the gradient-based Langevin dynamics of finite width neural networks with sufficiently small discretisation time step Alfonsi et al. (2015), as shown in Fig.1(e) and SI Sec.F . Even though in the diffusive phase, the dominant dy-

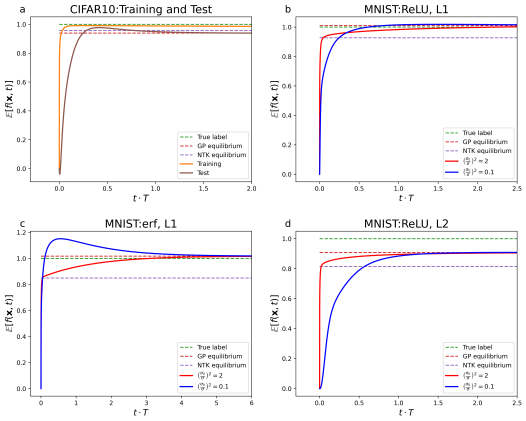

Figure 2: Dynamics of the mean predictor on a given test point in benchmark datasets. All test points shown have a target label $+1$. (a) Result on CIFAR10 dataset (Krizhevsky et al. (2014)) with binary classification of cats vs dogs, for $\sigma_0^2/\sigma^2 = 2$. We see a fast convergence of the mean predictor on the training point while the test point exhibits a diffusive learning phase on time scales $t \sim \mathcal{O}(1/T)$. (b-d) Results on MNIST dataset (Deng (2012)) with binary classification of 0 vs 1 digits, for $L = 1, 2$. In $L = 2$ the effect of σ_0^2/σ^2 is larger. (d) Results on MNIST dataset in a network with an error function (erf) nonlinearity with a single hidden layer. The effect σ_0^2/σ^2 is significantly larger than in (b,c).

namics is driven by noise and the regularization, the learning signal (both on the readout weights and the hidden layers) from the gradient of the loss is what restricts the exploration to the subspace of low ($\mathcal{O}(T)$) training error, and without it the performance will deteriorate back to chance.

The role of initialization and regularization and early stopping phenomena: We investigate how the diffusive dynamics is affected by the σ_0^2 for fixed values of σ^2 and T (thus fixing the time scale of the diffusive learning phase). As expected, the training predictor converges fast to the target output and exhibits little deviation afterward (see Fig.2 (a)). In the previous section, we kept the ratio σ_0^2/σ^2 fixed, resulting in the same qualitative behavior with different time scales. In Fig.2(b-d) , we show that changing the ratio σ_0^2/σ^2 results in qualitatively different behaviors of the trajectory, shown across network depth and nonlinearities. Interestingly, in Fig.2(c), when σ_0^2/σ^2 is small, the predictor dynamics is non-monotonic, overshooting above its equilibrium value. The optimal early stopping point, defined as the time the network reaches the minimal generalization error in the entire learning trajectory from $t = 0$ to $t \to \infty$, occurs in the diffusive learning phase. In this case, the performance in the diffusive phase is better than both equilibria. We study the effect of σ_0^2/σ^2 on the early stopping point systematically in the synthetic dataset in Fig.3.

The role of depth: The effect of different σ_0^2/σ^2 ratios on the dynamics increases with depth, resulting in distinctively different behavior for different ratios (Fig.2(b,d)). Depth also changes the NTK and NNGP equilibrium, typically in favor of the NNGP solution as the network grows deeper (see SI Sec.D.1). Furthermore, as shown in Fig.3, depth also has an effect on the occurrence of the optimal early stopping time. In the synthetic dataset, the early stopping time occurs earlier in shallower networks for small σ_0^2/σ^2, and does not occur when $L > 3$.

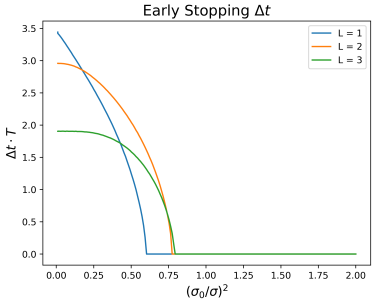

Figure 3: The time difference between the optimal stopping time and the long-time equilibrium time scaled by T (denoted by $\Delta t \cdot T$), for the synthetic orthogonal dataset in networks with hidden layers $L = 1, 2, 3$. We see that for small σ_0^2/σ^2 the optimal stopping time occurs during the diffusive learning phase, while for large σ_0^2/σ^2 the optimal stopping time is only at the long-time equilibrium, which corresponds to the NNGP. Interestingly, in this dataset for $L > 3$ there is no early stopping point.

The role of nonlinearity: We compare the behaviors of networks with ReLU and error function, with both having closed-form expressions for their NDK (see SI C.1-C.3). As shown in Fig.2(c) with error function nonlinearity, the difference between NTK and NNGP is larger and the effect of σ_0^2/σ^2 on the network dynamics is more significant.

5 Representational Drift

We now explore the implications of the diffusive learning dynamics on the phenomenon of representational drift. Representational drift refers to neuroscience observations of neuronal activity patterns

accumulating random changes over time without noticeable consequences on the relevant animal behavior. These observations raise fundamental questions about the causal relation between neuronal representations and the underlying computation. Some of these observations were in the context of learned behaviors and learning-induced changes in neuronal activity. One suggestion has been that changes in the readout of the circuit compensate for the representational drift, leaving intact its input-output relation (Rule et al. (2020); Rule & O'Leary (2022)). We provide a general theoretical framework for studying such dynamics. In our model, the stability of the (low) training error during the diffusion phase, is due to the continuous realignment of readout weights $\mathbf{a}(t)$ to changes in the network hidden layer weights $\mathbf{W}(t)$ as they drift simultaneously exploring the space of solutions.

The above alignment scenario requires an ongoing learning signal acting on the weights. To highlight the importance of this signal, we consider an alternative scenario where the readout weights are frozen at some time (denoted as t_0) after achieving a low training error while the weights of the hidden layers $\mathbf{W}(t)$ continue to drift randomly without an external learning signal. We will denote the output of the network in this scenario as $\mathrm{f}_{\text{drift}}(\mathbf{x}, t, t_0)$. Our formalism allows for computation of the mean of $\mathrm{f}_{\text{drift}}(\mathbf{x}, t, t_0)$ (see SI Sec.D for details). We present here the results for large t_0, i.e., after the learning has finished.

$$\mathbb{E}\left[\mathrm{f}_{\text{drift}}(t - t_0)\right] = \left(\mathbf{k}^L(t - t_0)\right)^\top \left(\boldsymbol{I} T\sigma^{-2} + \mathbf{K}_{GP}^L\right)^{-1} \boldsymbol{y} \tag{19}$$

The kernel $\mathbf{k}^L(t - t_0)$ represents the overlap between the representations of the training inputs at time t_0 and that of a test point at time t . When $t - t_0$ is large, the two representations completely decorrelate and the predictor is determined by a new kernel $K_{mean}^L(\mathbf{x}, \mathbf{x}')$defined as

$$K_{mean}^L(\mathbf{x}, \mathbf{x}') = N^{-1}\mathbb{E}_{\theta\sim\mathcal{N}(0,I\sigma^2)}\left[\mathbf{x}^L(\mathbf{x})\right] \cdot \mathbb{E}_{\theta\sim\mathcal{N}(0,I\sigma^2)}\left[\mathbf{x}^L(\mathbf{x}')\right] \tag{20}$$

which is a modified version of the NNGP kernel where the Gaussian averages are performed separately for each data point.

$$\lim_{t-t_0\to\infty} \mathbb{E}\left[\mathrm{f}_{\text{drift}}(\mathbf{x}, t - t_0)\right] = {\mathbf{k}_{mean}^L}^\top \left(\boldsymbol{I} T\sigma^{-2} + \mathbf{K}_{GP}^L\right)^{-1} \boldsymbol{y} \tag{21}$$

where $\mathbf{k}_{mean}^L$ is defined as applying the mean kernel function to the test data. For some nonlinearities (e.g., linear and error function activation) $K_{mean}^L(\mathbf{x}, \mathbf{x}')$ is zero. This however, is not the case for other nonlinearities (e.g., ReLU). In these cases, its value depends on the input vectors' norms $\|\mathbf{x}\|, \|\mathbf{x}'\|$. Thus, if the distribution of the norms is informative of the given task, the predictor can still be useful despite the drift process. In this case, we can say that the norms are drift-invariant information. In other cases, the norms may not be relevant to the task, in which case the decorrelated output will yield a chance-level performance. We present examples for both scenarios in Fig.4. We consider two MNIST binary classification tasks, after reaching the long-time equilibrium. For each one, we show the evolution of the histograms of the predictor on the training examples at times t, after freezing readout weights at an earlier time t_0. We train a linear classifier on top of the training predictors to evaluate the classification accuracy (see SI Sec.D for details). In the case of the classification task of the digit pair 4,9, the two histograms eventually overlap each other, resulting in a long-time chance level accuracy and a complete loss of the learned information. In contrast, in the classification of the digit pair 0,1 (Fig.4(f-j)), the histograms of the two classes are partially separated, leading to a long time accuracy of 90%, reflecting the residual information in the input norms. Interestingly during the dynamics from the original state to the long time state the distributions cross each other, resulting in a short period of chance performance.

6 Discussion

Our work provides the first theoretical understanding of the complete trajectory of gradient-descent learning dynamics of wide DNNs in the presence of small noise, unifying the NTK theory and the NNGP theory as two limits of the same underlying process. While the noise is externally injected in our setup, stochasticity in the machine-learning context may arise from randomness in the data in stochastic gradient descent, making noisy gradient descent a relevant setting in reality (Dalalyan (2017); Noh et al. (2017); Wu et al. (2020); Mignacco & Urbani (2022)). We derive a new kernel, the time-dependent NDK, as a dynamic generalization of the NTK, and provide new insights into learning dynamics in the diffusive learning phase as the learning process explores the solution space. We focus on two particularly interesting phenomena of early stopping and representational drift.

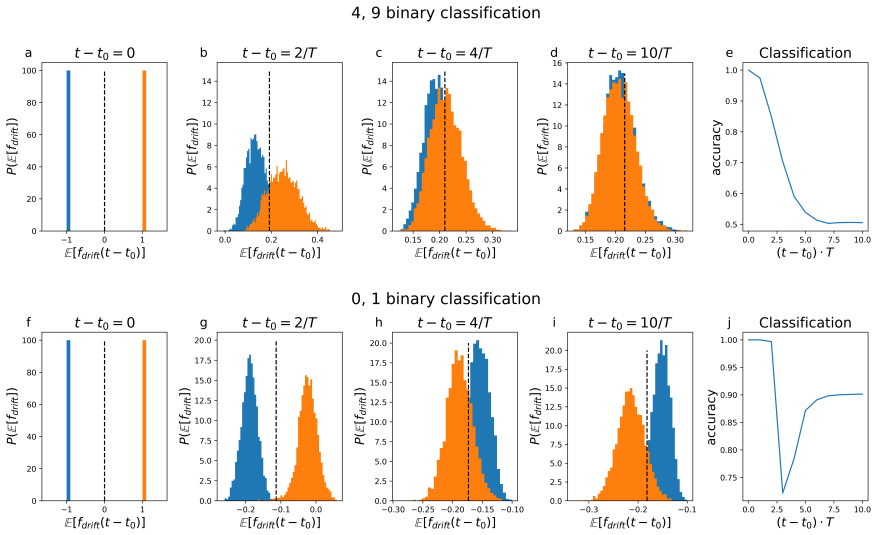

Figure 4: Representational drift with $\mathbf{a}_{t_0}$ fixed at a long-time equilibrium t_0. (a-d,f-i) The dynamics of the probability distribution of $\mathbb{E}\left[\mathrm{f}_{\mathrm{drift}}\left(\mathbf{x}, t - t_0\right)\right]$ over the training data, starting with two delta functions at ± 1, and gradually decays in performance when $\mathbf{a}\left(t_0\right)$ and $\mathbf{W}\left(t\right)$ lose alignment. On classification between the digits 0, 1, the norm of the images has enough information to classify them with reasonable success even after complete decorrelation, while on classification between the digits 4,9 the performance is reduced to chance. (e,j) The performance as a function of the time difference from the freezing point t_0.

We identify an important parameter σ_0^2/σ^2 characterizing the relative weights amplitude induced by initialization and Bayesian prior regularization, which plays an important role in shaping the trajectories of the predictor. We note that while the results are shown for network with a single output for simplicity, extension to networks with M outputs ($M \sim \mathcal{O}(1)$) is straightforward by simply replacing the P dimensional target output $\mathbf{y}$ to a $P \times M$ dimensional matrix.

In most of our examples, the best performance is achieved after the gradient-driven learning phase, indicating that exploring the solution space improves the network's performance, consistent with empirical findings (Lee et al. (2020)). For some examples, the optimal stopping point occurs during the diffusive phase, before the long-time equilibrium. We stress that our 'early stopping' is 'early' compared to the NNGP equilibrium, and is different from the usual notion of early stopping, which happens in the gradient-driven learning phase (Caruana et al. (2000); Jacot et al. (2018); Advani et al. (2020)). Our theory provides insights into how and when an early stopping point can happen after the network reaches an essentially zero training error.

Our theory for the Langevin dynamics suggests a possible mechanism of representational drift, where the hidden layer weights undergo random diffusion, while the readout weights are continuously realigning to keep performance unchanged, as previously suggested (Rule et al. (2020); Rule & O'Leary (2022)). In our framework, this realignment is due to the presence of a loss-gradient signal. The source of the putative realignment signals in brain circuits is unclear. An alternative hypothesis is that computations in the neuronal circuits are based on features that are invariant to the representational drift (Druckmann & Chklovskii (2012); Kaufman et al. (2014); Rule et al. (2019); Rubin et al. (2019); Deitch et al. (2021); Marks & Goard (2021)). We provide an example of such features and show that performance can be maintained after drift.

So far we have focused on learning in infinitely wide networks in the lazy regime, where the time dependence of the NDK results from random drift in the solution space. Empirical time-dependent NTK is more complex due to feature learning that exists in finite width NNs (Shan & Bordelon (2021); Vyas et al. (2022); Canatar & Pehlevan (2022)) or in an infinite width network with non-lazy regularization (Bordelon & Pehlevan (2022)). Future work aims to extend the theory to the regime where data size is proportional to network width where we expect dynamic kernel renormalization (Li & Sompolinsky (2021; 2022)) and to describe the dynamics of feature learning in non-lazy regularization (Woodworth et al. (2020); Azulay et al. (2021); Flesch et al. (2022)).

REFERENCES

Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

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.

Aurélien Alfonsi, Benjamin Jourdain, and Arturo Kohatsu-Higa. Optimal transport bounds between the time-marginals of a multidimensional diffusion and its euler scheme. 2015.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Shahar Azulay, Edward Moroshko, Mor Shpigel Nacson, Blake E Woodworth, Nathan Srebro, Amir Globerson, and Daniel Soudry. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In *International Conference on Machine Learning*, pp. 468–477. PMLR, 2021.

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

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.

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

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*, pp. 483–513. PMLR, 2020.

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

Abdulkadir Canatar and Cengiz Pehlevan. A kernel analysis of feature learning in deep neural networks. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1–8. IEEE, 2022.

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.

Rich Caruana, Steve Lawrence, and C Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. *Advances in neural information processing systems*, 13, 2000.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pp. 1305–1338. PMLR, 2020.

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

Arnak Dalalyan. Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent. In *Conference on Learning Theory*, pp. 678–689. PMLR, 2017.

Daniel Deitch, Alon Rubin, and Yaniv Ziv. Representational drift in the mouse visual cortex. *Current biology*, 31(19):4327–4339, 2021.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Shaul Druckmann and Dmitri B Chklovskii. Neuronal circuits underlying persistent representations despite time varying activity. *Current Biology*, 22(22):2095–2103, 2012.

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.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, New York, NY, 2nd edition, 2000. ISBN 978-0471056690.

Timo Flesch, Keno Juechems, Tsvetomira Dumbalska, Andrew Saxe, and Christopher Summerfield. Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, 110(7):1258–1270, 2022.

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

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.

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

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

Ziwei Ji, Justin Li, and Matus Telgarsky. Early-stopped neural networks are consistent. *Advances in Neural Information Processing Systems*, 34:1805–1817, 2021.

Hui Jin and Guido Montúfar. Implicit bias of gradient descent for mean squared error regression with wide neural networks. *arXiv preprint arXiv:2006.07356*, 2020.

Matthew T Kaufman, Mark M Churchland, Stephen I Ryu, and Krishna V Shenoy. Cortical activity in the null space: permitting preparation without movement. *Nature neuroscience*, 17(3):440–448, 2014.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5), 2014.

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.

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.

Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33:15156–15172, 2020.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pp. 4313–4324. PMLR, 2020.

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

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

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

Tyler D Marks and Michael J Goard. Stimulus-dependent representational drift in primary visual cortex. *Nature communications*, 12(1):5169, 2021.

Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

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.

Francesca Mignacco and Pierfrancesco Urbani. The effective noise of stochastic gradient descent. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(8):083405, 2022.

Hancheng Min, Salma Tarmoun, René Vidal, and Enrique Mallada. On the explicit role of initialization on the convergence and implicit bias of overparametrized linear networks. In *International Conference on Machine Learning*, pp. 7760–7768. PMLR, 2021.

Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, and Bohyung Han. Regularizing deep neural networks by noise: Its interpretation and optimization. *Advances in Neural Information Processing Systems*, 30, 2017.

Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A Alemi, Jascha Sohl-Dickstein, and Samuel S Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. *arXiv preprint arXiv:1912.02803*, 2019.

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

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

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

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

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

Alon Rubin, Liron Sheintuch, Noa Brande-Eilat, Or Pinchasof, Yoav Rechavi, Nitzan Geva, and Yaniv Ziv. Revealing neural correlates of behavior without behavioral measurements. *Nature communications*, 10(1):4745, 2019.

Michael E Rule and Timothy O'Leary. Self-healing codes: How stable neural populations can track continually reconfiguring neural representations. *Proceedings of the National Academy of Sciences*, 119(7):e2106692119, 2022.

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

Michael E Rule, Adrianna R Loback, Dhruva V Raman, Laura N Driscoll, Christopher D Harvey, and Timothy O'Leary. Stable task information from an unstable neural population. *Elife*, 9: e51121, 2020.

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

Carl E Schoonover, Sarah N Ohashi, Richard Axel, and Andrew JP Fink. Representational drift in primary olfactory cortex. *Nature*, 594(7864):541–546, 2021.

Haozhe Shan and Blake Bordelon. A theory of neural tangent kernel alignment and its influence on training. *arXiv preprint arXiv:2105.14301*, 2021.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Jascha Sohl-Dickstein, Roman Novak, Samuel S Schoenholz, and Jaehoon Lee. On the infinite width limit of neural networks with a standard parameterization. *arXiv preprint arXiv:2001.07301*, 2020.

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

Nikhil Vyas, Yamini Bansal, and Preetum Nakkiran. Limitations of the ntk for understanding generalization in deep learning. *arXiv preprint arXiv:2206.10012*, 2022.

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

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*, pp. 3635–3673. PMLR, 2020.

Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, Vladimir Braverman, and Zhanxing Zhu. On the noisy gradient descent that generalizes as sgd. In *International Conference on Machine Learning*, pp. 10367–10376. PMLR, 2020.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *Advances in Neural Information Processing Systems*, 32, 2019.

A MARKOV PROXIMAL LEARNING (MPL) FRAMEWORK FOR LEARNING DYNAMICS

We introduce a Markov proximal learning 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. We formally write down the moment generating function (MGF) of the predictor. We then use the well-known replica method in statistical physics (Mézard et al. (1987); Franz et al. (1992)), which has also been shown to be a powerful tool for deriving analytical results for learning in NNs (Gardner (1988); Gabrié et al. (2018); Carleo et al. (2019); Bahri et al. (2020); Saglietti & Zdeborová (2022)). 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.

A.1 DEFINITION OF MPL

We consider the network learning dynamics as a Markov proximal process, which is a generalized version of the *deterministic* proximal algorithm (Parikh et al. (2014); Polson et al. (2015)). Deterministic proximal algorithm with L_2 regularization is a sequential update rule defined as $\boldsymbol{\theta}_t\left(\boldsymbol{\theta}_{t-1}, \mathcal{D}\right) = \arg\min_{\boldsymbol{\theta}}\left(E\left(\boldsymbol{\theta}|\mathcal{D}\right) + \frac{\lambda}{2}\left|\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}\right|^2\right)$ where λ 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 (Teboulle (1997); Drusvyatskiy & Lewis (2018)), and many optimization algorithms widely used in machine learning can be seen as its approximations(Robbins & Monro (1951); Amari (1998); Beck & Teboulle (2003); Bae et al. (2022)). We define a stochastic extension of proximal learning, the Markov proximal learning, through the following transition matrix

$$\mathcal{T}\left(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}\right) = \frac{1}{Z\left(\boldsymbol{\theta}_{t-1}\right)}\exp\left(-\frac{1}{2}\beta\left(E\left(\boldsymbol{\theta}_t\right) + \frac{\lambda}{2}\left|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}\right|^2\right)\right) \tag{22}$$

where $Z\left(\boldsymbol{\theta}_{t-1}\right)$ is the single-time partition function, $Z\left(\boldsymbol{\theta}_{t-1}\right) = \int d\boldsymbol{\theta}' \mathcal{T}\left(\boldsymbol{\theta}'|\boldsymbol{\theta}_{t-1}\right)$. $\beta = T^{-1}$ is an inverse temperature parameter characterizing the level of 'uncertainty' and $\beta \to \infty$ limit recovers the deterministic proximal algorithm. We further assume that the initial distribution of $\boldsymbol{\theta}$ is an i.i.d. Gaussian with variance σ_0^2 and zero mean. Finally, we note that in the large λ limit, the difference between $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t-1}$ is infinitesimal, and $\boldsymbol{\theta}_t$ becomes a smooth function of continuous time, where the time variable is the discrete time divided by λ.

Large λ limit and Langevin dynamics:

We show that in the limit of large λ and differentiable cost function this algorithm is equivalent to gradient descent with white noise (Langevin dynamics). We define $\delta\boldsymbol{\theta}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}$. In the limit of large λ, we can expand the transition matrix around $\delta\boldsymbol{\theta}_t = 0$:

$$\mathcal{T}\left(\delta\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}\right) \approx \left(\frac{\lambda\beta}{4\pi}\right)^{\frac{d}{2}}\exp\left[-\frac{\lambda\beta}{4}\left|\delta\boldsymbol{\theta}_t + \frac{1}{\lambda}\nabla E\left(\boldsymbol{\theta}_{t-1}\right)\right|^2\right] \tag{23}$$

$\delta\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}$ is Gaussian with statistics:

$$\mathbb{E}\left[\delta\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}\right] = -\frac{1}{\lambda}\nabla E\left(\boldsymbol{\theta}_{t-1}\right) \tag{24}$$

$$\mathrm{Var}\left(\delta\boldsymbol{\theta}_t^i\delta\boldsymbol{\theta}_{t'}^j|\boldsymbol{\theta}_{t-1}\right) = \frac{2}{\lambda\beta}\delta_{ij}\delta_{t,t'} \tag{25}$$

which is equivalent to Langevin dynamics in Itô discretization:

$$\delta\boldsymbol{\theta}_t = \left(-\nabla E\left(\boldsymbol{\theta}_{t-1}\right) + \boldsymbol{\xi}_{t-1}\right)dt \tag{26}$$

with

$$\mathbb{E}\left[\boldsymbol{\xi}_t\boldsymbol{\xi}_{t'}^\top\right] = \frac{2T\boldsymbol{I}\delta_{t,t'}}{dt}, \mathbb{E}\left[\boldsymbol{\xi}_t\right] = 0 \tag{27}$$

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

B CALCULATION OF THE MOMENT GENERATION FUNCTION (MGF) AND THE MEAN PREDICTOR

In this section, we start from the MPL framework introduced in Section A.1, and present the detailed derivation of the moment generating function for the predictor statistics, explain the introduction of the auxilliary variables $\mathbf{v}(t)$ and $\mathbf{u}(t)$ in Eq.37, and derive expressions for the mean predictor given by Eq.15.

B.1 REPLICA CALCULATION OF THE MGF FOR THE PREDICTOR

The transition matrix can be written using the replica method, where

$$\begin{aligned} Z^{-1}\left(\boldsymbol{\theta}_{t-1}\right) &= \lim_{n\to 0} Z^{n-1}\left(\boldsymbol{\theta}_{t-1}\right) = \lim_{n\to 0}\left(\int d\boldsymbol{\theta}_t \exp\left(-\frac{\beta}{2}\left(E(\boldsymbol{\theta}_t)+\frac{\lambda}{2}\left|\boldsymbol{\theta}_t-\boldsymbol{\theta}_{t-1}\right|^2\right)\right)\right)^{n-1} \\ &= \lim_{n\to 0}\int \prod_{\alpha=1}^{n-1} d\boldsymbol{\theta}_t^\alpha \exp\left(-\frac{\beta}{2}\left(\sum_{\alpha=1}^{n-1} E(\boldsymbol{\theta}_t^\alpha)+\frac{\lambda}{2}\sum_{\alpha=1}^{n-1}\left|\boldsymbol{\theta}_t^\alpha-\boldsymbol{\theta}_{t-1}\right|^2\right)\right) \end{aligned} \tag{28}$$

therefore we have

$$\begin{aligned} \mathcal{T}\left(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}\right) = \mathcal{T}(\boldsymbol{\theta}_t^n|\boldsymbol{\theta}_{t-1}^n) &= \lim_{n\to 0} Z^{n-1}\left(\boldsymbol{\theta}_{t-1}^n\right)\exp\left(-\frac{1}{2}\beta\left(E\left(\boldsymbol{\theta}_t^n\right)+\frac{\lambda}{2}\left|\boldsymbol{\theta}_t^n-\boldsymbol{\theta}_{t-1}^n\right|^2\right)\right) \\ &= \lim_{n\to 0}\int \prod_{\alpha=1}^{n-1} d\boldsymbol{\theta}_t^\alpha \exp\left(-\frac{\beta}{2}\left(\sum_{\alpha=1}^{n} E\left(\boldsymbol{\theta}_t^\alpha\right)+\frac{\lambda}{2}\sum_{\alpha=1}^{n}\left|\boldsymbol{\theta}_t^\alpha-\boldsymbol{\theta}_{t-1}^n\right|^2\right)\right) \end{aligned} \tag{29}$$

Here $\alpha = 1,\cdots,n-1$ are the 'replicated copies' of the physical variable $\{\boldsymbol{\theta}_\tau^n\}_{\tau=1,\cdots,t} \equiv \{\boldsymbol{\theta}_\tau\}_{\tau=1,\cdots t}$. To calculate the statistics of the dynamical process, we consider the MGF for arbitrary functions of the trajectory $g(\{\boldsymbol{\theta}_\tau^n\}_{\tau=0,\cdots t})$, $\mathcal{M}\left[\ell_t\right] = \mathbb{E}\left[e^{\ell_t g\left(\{\boldsymbol{\theta}_\tau^n\}_{\tau=0\cdots t}\right)}\right]$

$$\mathcal{M}\left[\ell_t\right] = \prod_{\tau=0}^{\infty}\int d\boldsymbol{\theta}_\tau\left[\prod_{\tau=1}^{\infty}\mathcal{T}\left(\boldsymbol{\theta}_\tau|\boldsymbol{\theta}_{\tau-1}\right)\right]p\left(\boldsymbol{\theta}_0\right)\exp\left(\sum_{t=1}^{\infty}\ell_t g\left(\{\boldsymbol{\theta}_\tau^n\}_{\tau=0,\ldots t}\right)\right) \tag{30}$$

$$\begin{aligned} &= \lim_{n\to 0}\prod_{\alpha=1}^{n}\prod_{\tau=1}^{\infty}\int d\boldsymbol{\theta}_t^\alpha\int d\boldsymbol{\theta}_0^n p\left(\boldsymbol{\theta}_0^n\right) \\ &\exp\left(-\frac{\beta}{2}\sum_{\tau=1}^{\infty}\left(\sum_{\alpha=1}^{n}E\left(\boldsymbol{\theta}_\tau^\alpha\right)+\frac{\lambda}{2}\sum_{\alpha=1}^{n}\left|\boldsymbol{\theta}_\tau^\alpha-\boldsymbol{\theta}_{\tau-1}^n\right|^2\right)+\sum_{t=1}^{\infty}\ell_t g\left(\{\boldsymbol{\theta}_\tau^n\}_{\tau=0,\cdots t}\right)\right) \end{aligned} \tag{31}$$

We now apply this formalism to the cost function from Sec.2.1:

$$E\left(\boldsymbol{\theta}_t|\mathcal{D}\right) = \frac{1}{2}\left|\mathbf{f}_{\text{train}}\left(t\right)-\boldsymbol{y}\right|^2+\frac{T}{2\sigma^2}\left|\boldsymbol{\theta}_t\right|^2 \tag{32}$$

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

$$\mathcal{M}\left[\ell_t\right] = \lim_{n\to 0}\prod_{\alpha=1}^{n}\prod_{\tau=1}^{\infty}\int d\boldsymbol{\theta}_\tau^\alpha\int d\boldsymbol{\theta}_0\exp\left(-\frac{\beta}{4}\sum_{\tau=1}^{t}\sum_{\alpha=1}^{n}\left|\mathbf{f}_{\text{train}}\left(\boldsymbol{\theta}_t^\alpha\right)-\boldsymbol{y}\right|^2+\ell_t\mathbf{f}\left(\mathbf{x},\boldsymbol{\theta}_t^n\right)-S_0\left[\boldsymbol{\theta}\right]\right) \tag{33}$$

$$S_0\left[\boldsymbol{\theta}\right] = \frac{1}{4}\sum_{\tau=1}^{\infty}\sum_{\alpha=1}^{n}\left(\sigma^{-2}\left|\boldsymbol{\theta}_\tau^\alpha\right|^2+\lambda\beta\left|\boldsymbol{\theta}_\tau^\alpha-\boldsymbol{\theta}_{\tau-1}^n\right|^2\right)+\frac{1}{2}\sigma_0^{-2}\left|\boldsymbol{\theta}_0^n\right|^2 \tag{34}$$

where we define $\mathbf{f}_{\text{train}}\left(t\right) \equiv \left[\mathrm{f}\left(\mathbf{x}^1,t\right),\cdots,\mathrm{f}\left(\mathbf{x}^\mu,t\right)\right] \in \mathbb{R}^P$ a vector contains the predictor on the training dataset, and $\boldsymbol{y} \in \mathbb{R}^P$ as in the main text (Sec.2.1). $S_0\left(\boldsymbol{\theta}\right)$ denotes 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 $\mathbf{v}_\tau^\alpha \in \mathbb{R}^P$

$$\begin{aligned}\mathcal{M}\left[\ell_t\right] = \lim_{n\to 0} \prod_{\alpha=1}^{n} \prod_{\tau=1}^{\infty} \int d\boldsymbol{\theta}_\tau^\alpha \int d\mathbf{v}_\tau^\alpha \int d\boldsymbol{\theta}_0 \\ \exp\left(-\frac{i\beta}{2} \sum_{\tau=1}^{\infty} \sum_{\alpha=1}^{n} \left(\frac{1}{\sqrt{N_L}} \mathbf{f}_{\text{train}}\left(t\right) - \boldsymbol{y}\right)^\top \mathbf{v}_\tau^\alpha \right. \\ \left. -\frac{\beta}{4} \sum_{\tau=1}^{\infty} \sum_{\alpha=1}^{n} \left|\mathbf{v}_\tau^\alpha\right|^2 + \ell_t \mathrm{f}\left(\mathbf{x}, \boldsymbol{\theta}_t^n\right) - S_0\left(\boldsymbol{\theta}_\tau^\alpha\right)\right)\end{aligned} \tag{35}$$

Averaging over the readout weights a:

We denote the hidden layer weights collectively as $\mathcal{W}_\tau^\alpha = \left\{\mathbf{W}_\tau^{1,\alpha} \cdots \mathbf{W}_\tau^{L,\alpha}\right\}$. We integrate over $\mathbf{a}_\tau^\alpha$

$$\begin{aligned}\mathcal{M}\left[\ell_\tau\right] = \lim_{n\to 0} \prod_{\tau=1}^{\infty} \prod_{\alpha=1}^{n} \int d\mathbf{v}_\tau^\alpha \int d\mathcal{W}_\tau^\alpha \\ \exp\left(-S\left[\mathbf{v}_\tau^\alpha, \mathcal{W}_\tau^\alpha\right] - Q\left[\ell_t, \mathbf{v}_\tau^\alpha, \mathcal{W}_\tau^\alpha\right] - S_0\left[\mathcal{W}_\tau^\alpha\right]\right)\end{aligned} \tag{36}$$

$$S\left[\mathbf{v}_\tau^\alpha, \mathcal{W}_\tau^\alpha\right] = \frac{\beta}{4}\left(\sum_{\alpha,\beta=1}^{n} \sum_{\tau=1}^{\infty} \frac{\beta}{2} \mathbf{v}_\tau^{\alpha\top} m_{\tau,\tau'}^{\alpha\beta} \mathbf{K}_{\tau,\tau'}^{L,\alpha\beta}\left(\mathcal{W}_\tau^\alpha\right) \mathbf{v}_{\tau'}^\beta + \sum_{\alpha=1}^{n} \sum_{\tau=1}^{\infty} \left(\mathbf{v}_\tau^\alpha - 2iY\right)^\top \mathbf{v}_\tau^\alpha\right) \tag{37}$$

and the source term action

$$\begin{aligned}Q\left[\ell_t, \mathbf{v}_\tau^\alpha, \mathcal{W}_\tau^\alpha\right] =& i\frac{\beta}{2} \sum_{\alpha=1}^{n} \sum_{t,\tau=1}^{\infty} \mathbf{v}_\tau^{\alpha\top} m_{t,\tau}^{\alpha n} \mathbf{k}_{t,\tau}^{L,\alpha n}\left(\mathcal{W}_\tau^\alpha\right) \ell_t \\ &- \frac{1}{2} \sum_{t,t'=1}^{\infty} m_{t,t'}^{nn} k_{t,t'}^{L,nn}\left(\mathcal{W}_\tau^n\right) \ell_t \ell_{t'}\end{aligned} \tag{38}$$

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 $S_0\left[\boldsymbol{\theta}_\tau^\alpha\right]$, such that
$\mathbb{E}\left[\left(\boldsymbol{\theta}_\tau^\alpha\right)_i \left(\boldsymbol{\theta}_{\tau'}^\beta\right)_j\right]_{S_0} = \delta_{ij} m_{\tau,\tau'}^{\alpha\beta}$

$$m_{\tau,\tau'}^{\alpha\beta} = \begin{cases} m_{\tau,\tau'}^1 = \tilde{\sigma}^2\left(\tilde{\lambda}^{|\tau-\tau'|} + \gamma\tilde{\lambda}^{\tau+\tau'}\right) & \{\alpha=\beta, \tau=\tau'\} \cup \{\alpha=n, \tau<\tau'\} \cup \{\beta=n, \tau>\tau'\} \\ m_{\tau,\tau'}^0 = \tilde{\sigma}^2\left(\tilde{\lambda}^2\tilde{\lambda}^{|\tau-\tau'|} + \gamma\tilde{\lambda}^{\tau+\tau'}\right) & otherwise \end{cases} \tag{39}$$

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

$$\tilde{\lambda} = \frac{\lambda}{\lambda + T\sigma^{-2}}, \tilde{\sigma}^2 = \sigma^2 \frac{\lambda + T\sigma^{-2}}{\lambda + \frac{1}{2}T\sigma^{-2}}, \gamma = \frac{\sigma_0^2}{\tilde{\sigma}^2} - 1 \tag{40}$$

The time-dependent and replica-dependent kernel function $K_{\tau,\tau'}^{L,\alpha\beta}(\mathbf{x}, \mathbf{x}')$ is defined as:

$$K_{\tau,\tau'}^{L,\alpha\beta}\left(\mathbf{x}, \mathbf{x}'\right) = \frac{1}{N_L}\left(\mathbf{x}_\tau^L\left(\mathbf{x}, \mathcal{W}_\tau^\alpha\right) \cdot \mathbf{x}_{\tau'}^L\left(\mathbf{x}', \mathcal{W}_{\tau'}^\beta\right)\right) \tag{41}$$

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

Averaging over the hidden layer weights $\mathcal{W}$:

In the infinite width limit, the statistics of $\mathcal{W}$ are dominated by its Gaussian prior (Eq.34) with zero mean and covariance $\langle \mathcal{W}_\tau^\alpha \mathcal{W}_{\tau'}^{\beta\top} \rangle = m_{\tau,\tau'}^{\alpha\beta} \boldsymbol{I}$ .Thus the averaged kernel function $K_{\tau,\tau'}^{L,\alpha\beta}(\mathbf{x},\mathbf{x}')$ (Eq.41) over the prior yields two kinds of statistics for a given pair of time $\{\tau,\tau'\}$ as for $m_{\tau,\tau'}^{\alpha\beta}$, which we denote as $K_{\tau,\tau'}^{1,L}(\mathbf{x},\mathbf{x}')$, and $K_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}')$:

$$K_{\tau,\tau'}^{L,\alpha\beta}(\mathbf{x},\mathbf{x}') = \begin{cases} K_{\tau,\tau'}^{1,L}(\mathbf{x},\mathbf{x}') & \{\alpha=\beta,\tau=\tau'\}\cup\{\alpha=n,\tau<\tau'\}\cup\{\beta=n,\tau>\tau'\} \\ K_{\tau,\tau'}^{0}(\mathbf{x},\mathbf{x}') & otherwise \end{cases} \tag{42}$$

And they obey the iterative relations:

$$K_{\tau,\tau'}^{1,L}(\mathbf{x},\mathbf{x}') = F\left(m_{\tau,\tau}^{1}K_{\tau,\tau}^{1,L-1}(\mathbf{x},\mathbf{x}), m_{\tau',\tau'}^{1}K_{\tau',\tau'}^{1,L-1}(\mathbf{x}',\mathbf{x}'), m_{\tau,\tau'}^{1}K_{\tau,\tau'}^{1,L-1}(\mathbf{x},\mathbf{x}')\right) \tag{43}$$

$$K_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}') = F\left(m_{\tau,\tau}^{1}K_{\tau,\tau}^{1,L-1}(\mathbf{x},\mathbf{x}), m_{\tau',\tau'}^{1}K_{\tau',\tau'}^{1,L-1}(\mathbf{x}',\mathbf{x}'), m_{\tau,\tau'}^{0}K_{\tau,\tau'}^{0,L-1}(\mathbf{x},\mathbf{x}')\right) \tag{44}$$

$$K^{1,L=0}(\mathbf{x},\mathbf{x}') = K^{0,L=0}(\mathbf{x},\mathbf{x}') = K^{in}(\mathbf{x},\mathbf{x}') \tag{45}$$

$$K^{in}(\mathbf{x},\mathbf{x}') = \frac{1}{N_0}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i' \tag{46}$$

where $F\left(\mathbb{E}\left[z^2\right],\mathbb{E}\left[z'^2\right],\mathbb{E}\left[zz'\right]\right)$ 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 (Cho & Saul (2009)). As we see in Eqs.43,44 these variances and covariances depend on the kernel functions of the previous layer and on the prior replica-dependent 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 $\Delta_{\tau,\tau'}^{L}(\mathbf{x},\mathbf{x}') = \frac{\lambda\beta}{2}\left(K_{\tau,\tau'}^{1,L}(\mathbf{x},\mathbf{x}') - K_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}')\right)$. It is useful to define a new kernel, the discrete neural dynamical kernel $K_{\tau,\tau'}^{d,L}(\mathbf{x},\mathbf{x}') = \lim_{n\to 0}\frac{\lambda\beta}{2}\sum_{\alpha=1}^{n} m_{\tau,\tau'}^{n\beta}K_{\tau,\tau'}^{n\beta,L}(\mathbf{x},\mathbf{x}')$, which controls the dynamics of the mean predictor. It has a simple expression in terms of the kernel $K_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}')$ and the kernel difference $\Delta_{\tau,\tau'}^{L}$.

$$K_{\tau,\tau'}^{d,L}(\mathbf{x},\mathbf{x}') = \begin{cases} 0 & \tau\le\tau' \\ m_{\tau,\tau'}^{1}\Delta_{\tau,\tau'}^{L}(\mathbf{x},\mathbf{x}') + \tilde{\lambda}^{|\tau-\tau'|+1}K_{\tau,\tau'}^{0,L}(\mathbf{x},\mathbf{x}') & \tau>\tau' \end{cases} \tag{47}$$

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

$$\mathcal{M}[\ell_t] = \lim_{n\to 0}\prod_{\alpha=1}^{n}\prod_{\tau=1}^{\infty}\int d\mathbf{v}_\tau^\alpha \exp\left(-S(\mathbf{v}_\tau^\alpha) - Q(\ell_t,\mathbf{v}_\tau^\alpha)\right) \tag{48}$$

$$\begin{aligned} S[\mathbf{v}_\tau^\alpha] = \frac{\beta}{4}\sum_{\tau=1}^{\infty}\Bigg(&\frac{\beta}{2}\sum_{\alpha,\beta=1}^{n}\sum_{\tau'=1}^{\infty}\mathbf{v}_\tau^{\alpha\top}m_{\tau,\tau'}^{0}\mathbf{K}_{\tau,\tau'}^{0,L}\mathbf{v}_{\tau'}^{\beta} + \frac{2}{\lambda}\sum_{\alpha=1}^{n}\sum_{\tau'=1}^{t-1}\mathbf{v}_\tau^{\alpha\top}\mathbf{K}_{\tau,\tau'}^{d,L}\mathbf{v}_{\tau'}^{n} \\ &+\frac{1}{\lambda}\sum_{\alpha=1}^{n}\mathbf{v}_\tau^{\alpha\top}\mathbf{K}_{\tau,\tau}^{d,L}\mathbf{v}_\tau^{\alpha} + \sum_{\alpha=1}^{n}\mathbf{v}_\tau^{\alpha\top}\left(\mathbf{v}_\tau^{\alpha} - 2i\boldsymbol{y}\right)\Bigg) \end{aligned} \tag{49}$$

$$\begin{aligned} Q[\ell_t,\mathbf{v}_\tau^\alpha] =& \frac{i\beta}{2}\sum_{\beta=1}^{n}\sum_{t,\tau'=1}^{\infty}\ell_t m_{t,\tau'}^{0}\mathbf{k}_{t,\tau'}^{0,L\top}\mathbf{v}_{\tau'}^{\beta} + \frac{i}{\lambda}\sum_{t,\tau'=1}^{t}\ell_t\mathbf{k}_{t,\tau'}^{d,L\top}\mathbf{v}_{\tau'}^{n} \\ &+\frac{i}{\lambda}\sum_{\beta=1}^{n}\sum_{t=1}^{\infty}\sum_{\tau'=t+1}^{\infty}\ell_t\mathbf{k}_{t,\tau'}^{d,L\top}\mathbf{v}_{\tau'}^{\beta} - \sum_{t=1}^{\infty}\frac{1}{2}m_{t,t}^{1}\ell_t^2k_{t,t}^{1,L} \end{aligned} \tag{50}$$

$\mathbf{k}_{t,\tau'}^{d,L}$ in Eq.50 is a P-dimensional vector given by applying the kernel function on the test data.

B.2 INTEGRATE OUT REPLICATED VARIABLES $\mathbf{v}_\tau^\alpha$

We define a new variable $\mathbf{u}_\tau = \frac{\lambda\beta}{2}\sum_{\alpha=1}^{n}\mathbf{v}_\tau^\alpha$, and integrate out $\mathbf{v}_\tau^{\alpha\neq n}$, we obtain a simpler expression of the MGF (after taking the limit $n \to 0$).

$$\mathcal{M}\left[\ell_t\right] = \prod_{\tau=1}^{\infty}\int d\mathbf{v}_\tau \int d\mathbf{u}_\tau \exp\left(-S\left[\mathbf{v}_\tau, \mathbf{u}_\tau\right] - Q\left[\ell_\tau, \mathbf{v}_\tau, \mathbf{u}_\tau\right]\right) \tag{51}$$

$$\begin{aligned} S\left[\mathbf{v}_\tau, \mathbf{u}_\tau\right] = &\frac{1}{2\lambda^2}\sum_{\tau,\tau'=1}^{\infty}\mathbf{u}_\tau^\top\left(m_{\tau,\tau'}^0\mathbf{K}_{\tau,\tau'}^{0,L} - \frac{2}{\beta}\delta_{\tau,\tau'}\left(\boldsymbol{I} + \frac{1}{\lambda}\mathbf{K}_{\tau,\tau}^{d,L}\right)\right)\mathbf{u}_{\tau'} \\ &+ \frac{1}{\lambda}\sum_{\tau=1}^{\infty}\left(\frac{1}{\lambda}\sum_{\tau'=1}^{\tau-1}\mathbf{K}_{\tau,\tau'}^{d,L}\mathbf{v}_{\tau'} + \left(\boldsymbol{I} + \frac{1}{\lambda}\mathbf{K}_{\tau,\tau}^{d,L}\right)\mathbf{v}_\tau - i\boldsymbol{y}\right)^\top\mathbf{u}_\tau \end{aligned} \tag{52}$$

$$\begin{aligned} Q\left[\ell_\tau, \mathbf{v}_\tau, \mathbf{u}_\tau\right] = &\frac{i}{\lambda}\sum_{t=1}^{\infty}\ell_t\left(\sum_{\tau'=1}^{\infty}m_{t,\tau'}^0\mathbf{k}_{t,\tau'}^{0,L\top}\mathbf{u}_{\tau'} + \sum_{\tau'=1}^{t}\mathbf{k}_{t,\tau'}^{d,L\top}\mathbf{v}_{\tau'} + \frac{2}{\lambda\beta}\sum_{\tau'=t+1}^{\infty}\mathbf{k}_{t,\tau'}^{d,L\top}\mathbf{u}_{\tau'}\right) \\ &- \sum_{t=1}^{\infty}\frac{1}{2}\left(\ell_t\right)^2 m_{t,t}^1 k_{t,t}^{1,L} \end{aligned} \tag{53}$$

B.3 DETAILED CALCULATION OF THE MEAN PREDICTOR

To derive the mean predictor we take the derivative of the MGF w.r.t. ℓ_t:

$$\mathbb{E}\left[\mathrm{f}\left(\mathbf{x}, t\right)\right] = \left.\frac{\partial\mathcal{M}\left(\ell_t\right)}{\partial\ell_t}\right|_{\ell_t=0} \tag{54}$$

which yields

$$\mathbb{E}\left[\mathrm{f}\left(\mathbf{x}, t\right)\right] = \frac{1}{\lambda}\sum_{t'=1}^{t}\mathbf{k}_{t,t'}^{d,L\top}\mathbb{E}\left[-i\mathbf{v}_{t'}\right] \tag{55}$$

Furthermore, from the H.S. transformation in Eq.35, we can relate $\mathbb{E}\left[\mathbf{v}_\tau\right]$ to the mean predictor on the training data :

$$\mathbb{E}\left[i\mathbf{v}_t\right] = \mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t\right)\right] - \boldsymbol{y} \tag{56}$$

On the other hand we can get the statistics of $i\mathbf{v}_t$ from the MGF in Eq.51.

$$\mathbb{E}\left[\left(\mathbf{f}_{\text{train}}\right)_t\right] = \left(\boldsymbol{I}\lambda + \mathbf{K}_{t,t}^{d,L}\right)^{-1}\sum_{t'=1}^{t-1}\mathbf{K}_{t,t'}^{d,L}\left(\boldsymbol{y} - \mathbb{E}\left[\left(\mathbf{f}_{\text{train}}\right)_{t'}\right]\right) \tag{57}$$

$$\mathbb{E}\left[\mathrm{f}\left(\mathbf{x}, t\right)\right] = \frac{1}{\lambda}\sum_{t'=1}^{t}\mathbf{k}_{t,t'}^{d,L\top}\left(\boldsymbol{y} - \mathbb{E}\left[\left(\mathbf{f}_{\text{train}}\right)_{t'}\right]\right) \tag{58}$$

where $\mathbf{K}^{d,L}\left(t, t'\right)$ is a $P \times P$ dimensional kernel matrix defined as $\mathbf{K}_{\mu\nu,t,t'}^{d,L} = \mathrm{K}_{t,t'}^{d,L}\left(\mathbf{x}^\mu, \mathbf{x}^\nu\right)$. Now we can compute $\mathbb{E}\left[\mathrm{f}\left(\mathbf{x}, \boldsymbol{\theta}_t\right)\right]$ iteratively by combining Eqs.57,58.

B.4 LARGE λ LIMIT

All the results so far hold for any T and λ. 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 gradient-descent scenario. We consider large λ and $t_{discrete} \sim O\left(\lambda\right)$, and thus define a new continuous-time $t = t_{discrete}/\lambda \sim O\left(1\right)$. In this limit, the parameters defined in Eq.40 become

$$\tilde{\lambda}^{t_{discrete}} = e^{-T\sigma^{-2}t}, \tilde{\sigma}^2 = \sigma^2 \tag{59}$$

Taking the limit of large λ of Eq.51 is straightforward, and yields

$$\mathcal{M}\left[\ell\left(t\right)\right]=\int D\mathbf{v}\left(t\right)\int D\mathbf{u}\left(t\right)\exp\left(-S\left[\mathbf{v}\left(t\right),\mathbf{u}\left(t\right)\right]-Q\left[\ell\left(t\right),\mathbf{v}\left(t\right),\mathbf{u}\left(t\right)\right]\right) \tag{60}$$

where

$$\begin{aligned}S\left[\mathbf{v}\left(t\right),\mathbf{u}\left(t\right)\right]=&\frac{1}{2}\int_{0}^{\infty}dt\int_{0}^{\infty}dt'm\left(t,t'\right)\mathbf{u}^{\top}\left(t\right)\mathbf{K}^{L}\left(t,t'\right)\mathbf{u}\left(t'\right)\\&+\int_{0}^{\infty}dt\left(\int_{0}^{t}dt'\mathbf{K}^{d,L}\left(t,t'\right)\mathbf{v}\left(t'\right)+\mathbf{v}\left(t\right)-i\boldsymbol{y}\right)^{\top}\mathbf{u}\left(t\right)\end{aligned} \tag{61}$$

and the source term action is

$$\begin{aligned}Q\left[\ell\left(t\right),\mathbf{v}\left(t\right),\mathbf{u}\left(t\right)\right]=&i\int_{0}^{\infty}dt\int_{0}^{t}dt'\left(\mathbf{k}^{d,L}\left(t,t'\right)\right)^{\top}\mathbf{v}\left(t'\right)\ell\left(t\right)\\&+i\int_{0}^{\infty}dt\int_{0}^{\infty}dt'm\left(t,t'\right)\left(\mathbf{k}^{L}\left(t,t'\right)\right)^{\top}\mathbf{u}\left(t'\right)\ell\left(t\right)\\&-\frac{1}{2}\int_{0}^{\infty}dt\int_{0}^{\infty}dt'm\left(t,t'\right)k^{L}\left(t,t'\right)\ell\left(t\right)\ell\left(t'\right)\end{aligned} \tag{62}$$

The NDK in Eq.47 can be rewritten as

$$K^{d,L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=m\left(t,t'\right)\Delta^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)+e^{-T\sigma^{-2}\left|t-t'\right|}K^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right) \tag{63}$$

with

$$\begin{aligned}\Delta^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=&\frac{\lambda}{2T}\left(K^{L,1}\left(t,t',\mathbf{x},\mathbf{x}'\right)-K^{L,0}\left(t,t',\mathbf{x},\mathbf{x}'\right)\right)\\=&K^{d,L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\dot{K}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)\end{aligned} \tag{64}$$

$$m\left(t,t'\right)=\sigma^{2}e^{-T\sigma^{-2}\left|t-t'\right|}+\left(\sigma_{0}^{2}-\sigma^{2}\right)e^{-T\sigma^{-2}\left(t+t'\right)} \tag{65}$$

with the kernels defined in Sec.2.3 in the main text. Here the quantity $m\left(t,t'\right)$ is the continuous time limit of $m_{t,t'}^{1}$. As defined in Eq.39, it represents the covariance of the prior

$$\mathbb{E}\left[\boldsymbol{\theta}_{t}^{i}\boldsymbol{\theta}_{t'}^{j}\right]_{S_{0}}=\delta_{ij}m\left(t,t'\right),\mathbb{E}\left[\boldsymbol{\theta}_{t}^{i}\right]_{S_{0}}=0 \tag{66}$$

.

The above calculation leads to the recursion relation of $\mathrm{K}^{d,L}\left(t,t',\mathbf{x},\mathbf{x}'\right)$ given in Eq.12 in the main text:

$$\begin{aligned}K^{d,L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=&m\left(t,t'\right)\mathrm{K}^{d,L-1}\left(t,t',\mathbf{x},\mathbf{x}'\right)\dot{K}^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)\\&+e^{-T\sigma^{-2}\left|t-t'\right|}K^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right)\end{aligned} \tag{67}$$

with initial condition

$$K^{d,L=0}\left(t,t',\mathbf{x},\mathbf{x}'\right)=e^{-T\sigma^{-2}\left|t-t'\right|}K^{in}\left(\mathbf{x},\mathbf{x}'\right) \tag{68}$$

Where $K^{in}\left(\mathbf{x},\mathbf{x}'\right)$ was defined in Eq.46. We refer to this continuous time $\mathrm{K}^{d,L}\left(t,t',\mathbf{x},\mathbf{x}'\right)$ as the neural dynamical kernel (NDK). Note that it follows directly from Eq.67 that

$$K^{d,L}\left(0,0,\mathbf{x},\mathbf{x}'\right)=K_{NTK}^{L}\left(\mathbf{x},\mathbf{x}'\right). \tag{69}$$

For the mean predictor we use the results from the previous section Eqs.56,57,58, take the large λ limit and turn the sums into integrals, we obtain

$$\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t\right)\right]=\int_{0}^{t}dt'\mathbf{K}^{d,L}\left(t,t'\right)\left(\boldsymbol{y}-\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t'\right)\right]\right) \tag{70}$$

$$\mathbb{E}\left[\mathrm{f}\left(\mathbf{x},t\right)\right]=\int_{0}^{t}dt'\left(\mathbf{k}^{d,L}\left(t,t'\right)\right)^{\top}\left(\boldsymbol{y}-\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t'\right)\right]\right) \tag{71}$$

as given in Eqs.14, 15 in the main text.

B.5 Temporal Correlations

Previously we considered the predictor with readout weights $\mathbf{a}_t$ and hidden layer weights $\mathcal{W}_t$ at the same time t. To reveal the effects of learning on $\mathcal{W}$ and $\mathbf{a}$ separately, we can consider the temporal correlation between $\mathcal{W}$ and $\mathbf{a}$ at different times:

$$c\left(\mathbf{x},t_0,t\right)\equiv\mathbb{E}\left[\mathrm{f}\left(\mathbf{x},\mathbf{a}_{t_0},\mathcal{W}_t\right)\right]=\mathbb{E}\left[\frac{1}{\sqrt{N_L}}\mathbf{a}_{t_0}\cdot\mathbf{x}_t^L\left(\mathbf{x},\mathcal{W}_t\right)\right] \tag{72}$$

We can derive the MGF of this quantity by replacing $\ell g\left(\left\{\boldsymbol{\theta}_\tau^n\right\}_{\tau=1,\cdots,t}\right)$ in Eq.31 by $\ell\left(t_0,t\right)c\left(t_0,t\right)$. For convenience, we split the action into three parts, one that previously appeared in the equal time calculation in Eq.61, and two new parts involving the new source $\ell\left(t_0,t\right)$.

$$\begin{aligned}\mathcal{M}\left[\ell\left(t_0,t\right)\right]=\int D\mathbf{v}\left(t\right)\int D\mathbf{u}\left(t\right)\exp\left(-S\left[\mathbf{v}\left(t\right),\mathbf{u}\left(t\right)\right]\right.\\ \left.-Q_1\left[\ell\left(t_0,t\right),\mathbf{u}\left(t\right)\right]-Q_2\left[\ell\left(t_0,t\right),\mathbf{v}\left(t\right)\right]\right)\end{aligned} \tag{73}$$

$$\begin{aligned}Q_1\left[\ell\left(t_0,t\right),\mathbf{u}\left(t\right)\right]=&\int_0^\infty dt_0\int_0^\infty dt\int_0^\infty dt'm\left(t_0,t'\right)\left(\mathbf{k}^L\left(t,t'\right)\right)^{\top}\mathbf{u}\left(t'\right)\ell\left(t_0,t\right)\\&+\frac{1}{2}\int_0^\infty dt\int_0^\infty dt'\int_0^\infty dt_0\int_0^\infty dt_0'm\left(t_0,t_0'\right)k^L\left(t,t'\right)\ell\left(t_0,t\right)\ell\left(t_0',t'\right)\end{aligned} \tag{74}$$

$$\begin{aligned}Q_2\left[\ell\left(t_0,t\right),\mathbf{v}\left(t\right)\right]=&\int_0^\infty dt\int_0^\infty dt_0\int_0^{\max(t_0,t)}dt'\ell\left(t_0,t\right)\mathbf{v}^{\top}\left(t'\right)\\&\left(\boldsymbol{\theta}\left(t-t'\right)m\left(t_0,t'\right)\mathbf{k}^{d,L-1}\left(t,t'\right)\dot{\mathbf{k}}^L\left(t,t'\right)\right.\\&\left.+\boldsymbol{\theta}\left(t_0-t'\right)e^{-T\sigma^{-2}\left|t_0-t'\right|}\mathbf{k}^L\left(t,t'\right)\right)\end{aligned} \tag{75}$$

Using the same approach as in Sec.B.3, we get the statistics of $c\left(\mathbf{x},t_0,t\right)$, which depend on whether $t>t_0$ or vice versa:

$$\begin{aligned}c\left(\mathbf{x},t_0<t\right)=&e^{T\sigma^{-2}\left(t-t_0\right)}\int_0^{t_0}dt'\left(\mathbf{k}^{d,L}\left(t,t'\right)\right)^{\top}\left(\boldsymbol{y}-\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t'\right)\right]\right)\\&+\int_{t_0}^{t}dt'm\left(t',t_0\right)\left(\mathbf{k}^{d,L-1}\left(t,t'\right)\dot{\mathbf{k}}^L\left(t,t'\right)\right)^{\top}\left(\boldsymbol{y}-\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t'\right)\right]\right)\end{aligned} \tag{76}$$

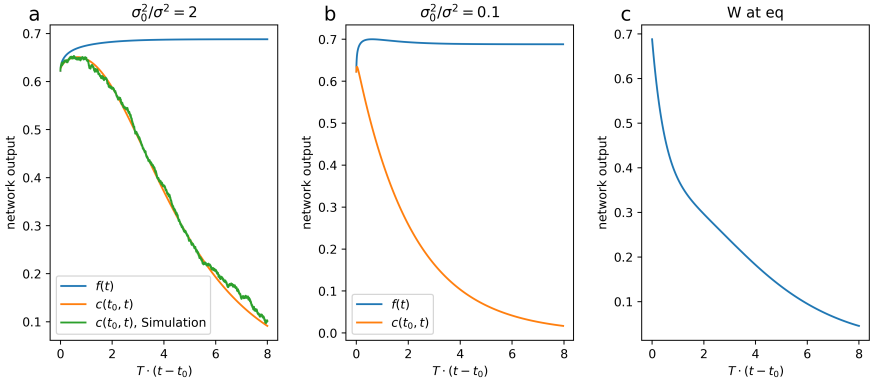

Figure 5: Temporal correlation dynamics for the synthetic dataset. (a-b) Temporal correlations between $\mathbf{a}_{t_0}$ fixed at NTK equilibrium with changing $\mathcal{W}_t$, for different σ_0^2/σ^2 values. In (a), we show remarkable agreement between the theory and network simulations. Interestingly, for larger σ_0^2/σ^2, the temporal correlations closely follow the mean predictor dynamics, meaning the learning of $\mathcal{W}_t$ is dominant in this regime. (b) There is almost exponential decay of the temporal correlations, similar to Fig.11, meaning the effect of learning on $\mathcal{W}_t$ is weak (almost like the representational drift case). (c) $\mathcal{W}_t$ fixed at NNGP equilibrium, while the dynamics of $\mathbf{a}_{t_0}$ continues.

$$c\left(\mathbf{x}, t_0 > t\right) = e^{-T\sigma^{-2}(t_0 - t)} \mathbb{E}\left[\mathrm{f}\left(\mathbf{x}, t\right)\right] + \int_t^{t_0} dt' e^{-T\sigma^{-2}(t_0 - t)} \left(\mathbf{k}^L\left(t, t'\right)\right)^\top \left(\boldsymbol{y} - \mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t'\right)\right]\right) \tag{77}$$

The kernels are defined in Sec.2.3. $\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t\right)\right]$ is calculated via the integral equation in Eq.14 in the main text. By definition $c\left(\mathbf{x}, t = t_0\right) = \mathbb{E}\left[\mathrm{f}\left(\mathbf{x}, t\right)\right]$.

Solving the integrals numerically, we find the the ratio σ_0^2/σ^2 plays an important role in the dynamics again. As can be seen in Fig.5 (a), when σ_0^2/σ^2 is large, the temporal correlations follow the predictor for a significant amount of time even though $\mathbf{a}_{t_0}$ is frozen, meaning that the effect of learning on the hidden layer weights $\mathcal{W}_t$ is dominant. Eventually, the decorrelation between $\mathbf{a}_{t_0}$ and $\mathcal{W}_t$ causes a decrease in performance. When σ_0^2/σ^2 is small (Fig.5 (b)), the temporal correlations decrease almost exponentially, hinting that in this regime the effect of learning on the readout weights is dominant. In this case Fig.5 (b) is similar to Fig.11, where there is no external learning signal affecting the hidden layer weights at all.

C THE NEURAL DYNAMICAL KERNEL

We focus on the large λ 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.

C.1 LINEAR ACTIVATION:

For linear activation:

$$K^L\left(t, t', \mathbf{x}, \mathbf{x}'\right) = \left(m\left(t, t'\right)\right)^L K^{in}\left(\mathbf{x}, \mathbf{x}'\right) \tag{78}$$

$$\dot{K}^L\left(t, t', \mathbf{x}, \mathbf{x}'\right) = \boldsymbol{I} \tag{79}$$

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

$$K^{d,L}\left(t, t', \mathbf{x}, \mathbf{x}'\right) = \left(m\left(t, t'\right)\right)^L \left(L + 1\right) e^{-T\sigma^{-2}\left|t - t'\right|} K^{in}\left(\mathbf{x}, \mathbf{x}'\right) \tag{80}$$

The NDK of linear activation is proportional to the input kernel $K^{in}(\mathbf{x},\mathbf{x}')$ 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 $\mathcal{O}(T)$. This suggests that the diffusive phase has very little effect on the mean predictor in the low T regime, as shown in Fig.10.

C.2 ReLU activation:

For ReLU activation, we define the function $J(\boldsymbol{\theta})$ (Cho & Saul (2009)):

$$J\left(\boldsymbol{\theta}^L(t,t',\mathbf{x},\mathbf{x}')\right) = \left(\pi - \boldsymbol{\theta}^L(t,t',\mathbf{x},\mathbf{x}')\right)\cos\left(\boldsymbol{\theta}^L(t,t',\mathbf{x},\mathbf{x}')\right) + \sin\left(\boldsymbol{\theta}^L(t,t',\mathbf{x},\mathbf{x}')\right) \tag{81}$$

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

$$\boldsymbol{\theta}^L(t,t',\mathbf{x},\mathbf{x}') = \cos^{-1}\left(\frac{m(t,t')}{\sqrt{m(t,t)\,m(t',t')}}\frac{1}{\pi}J\left(\boldsymbol{\theta}^{L-1}(t,t',\mathbf{x},\mathbf{x}')\right)\right) \tag{82}$$

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

$$\boldsymbol{\theta}^{L=0}(t,t',\mathbf{x},\mathbf{x}') = \cos^{-1}\left(\frac{m(t,t')}{\sqrt{m(t,t)\,m(t',t')}}\frac{K^{in}(\mathbf{x},\mathbf{x}')}{\sqrt{K^{in}(\mathbf{x},\mathbf{x})K^{in}(\mathbf{x}',\mathbf{x}')}}\right) \tag{83}$$

the kernel functions are then given by

$$\dot{K}^L(t,t',\mathbf{x},\mathbf{x}') = \frac{1}{2\pi}\left(\pi - \boldsymbol{\theta}^L(t,t',\mathbf{x},\mathbf{x}')\right) \tag{84}$$

$$K^L(t,t',\mathbf{x},\mathbf{x}') = \frac{\sqrt{K^{in}(\mathbf{x},\mathbf{x})\,K^{in}(\mathbf{x}',\mathbf{x}')}}{\pi 2^L}\left(m(t,t)\,m(t',t')\right)^{L/2}J\left(\boldsymbol{\theta}^{L-1}(t,t',\mathbf{x},\mathbf{x}')\right) \tag{85}$$

We obtain an explicit expression for the NDK by plugging these kernels into Eqs.67,68.

C.3 Error function activation

For error function activation (Williams (1996)):

$$\begin{aligned} &K^L(t,t',\mathbf{x},\mathbf{x}') \\ &= \frac{2}{\pi}\sin^{-1}\left(\frac{2m(t,t')\,K^{L-1}(t,t',\mathbf{x},\mathbf{x}')}{\sqrt{\left(1+2m(t,t)\,K^{L-1}(t,t,\mathbf{x},\mathbf{x})\right)\left(1+2m(t',t')\,K^{L-1}(t',t',\mathbf{x}',\mathbf{x}')\right)}}\right) \end{aligned} \tag{86}$$

$$\begin{aligned} \dot{K}^L_{\mu\nu}(t,t',\mathbf{x},\mathbf{x}') =& \frac{4}{\pi}\Big(\left(1+2m(t,t)\,K^{L-1}(t,t,\mathbf{x},\mathbf{x})\right)\left(1+2m(t',t')\,K^{L-1}(t',t',\mathbf{x}',\mathbf{x}')\right) \\ &-4\left(m(t,t')\,K^{L-1}(t,t',\mathbf{x},\mathbf{x}')\right)^2\Big)^{-1/2} \end{aligned} \tag{87}$$

Again we can obtain an explicit expression for the NDK by plugging these kernels into Eqs.67,68.

C.4 Long-time behavior of the NDK

We define the long-time limit as $t,t'\to\infty, t-t'\sim\mathcal{O}\left(T^{-1}\right)$. In this limit the statistics of $\mathcal{W}$ w.r.t. the prior becomes only a function of the time difference:

$$\mathbb{E}\left[\mathcal{W}_t\mathcal{W}_{t'}^\top\right] = \sigma^2 e^{-T\sigma^{-2}|t-t'|} = m(|t-t'|) \tag{88}$$

And thus, the kernels defined above will only be functions of the time difference. We look at the time derivative of the kernel (w.l.o.g. we assume $t>t'$), which can be obtained with a chain rule:

$$\frac{d}{dt'}K^L(t-t',\mathbf{x},\mathbf{x}') = \dot{K}^L(t-t',\mathbf{x},\mathbf{x}')\frac{d}{dt'}\left(K^{L-1}(t-t',\mathbf{x},\mathbf{x}')\,m(t-t')\right) \tag{89}$$

We prove by induction:

$$\frac{1}{T}\frac{d}{dt'}\left(m\left(t-t'\right)K^{L}\left(t-t',\mathbf{x},\mathbf{x}'\right)\right)=K^{d,L}\left(t-t',\mathbf{x},\mathbf{x}'\right) \tag{90}$$

The induction basis for $L=0$ is trivial. For arbitrary $L+1$:

$$\begin{aligned}&\frac{1}{T}\frac{d}{dt'}\left(m\left(t-t'\right)K^{L+1}\left(t-t',\mathbf{x},\mathbf{x}'\right)\right) &(91)\\ &\quad=m\left(t-t'\right)\dot{K}^{L+1}\left(t-t',\mathbf{x},\mathbf{x}'\right)\frac{1}{T}\frac{d}{dt'}\left(K^{L}\left(t-t',\mathbf{x},\mathbf{x}'\right)m\left(t-t'\right)\right)\\ &\quad+e^{-T\sigma^{-2}\left(t-t'\right)}K^{L+1}\left(t-t',\mathbf{x},\mathbf{x}'\right) &(92)\end{aligned}$$

And using the induction assumption we get:

$$\begin{aligned}\frac{1}{T}\frac{d}{dt'}\left(m\left(t-t'\right)K^{L+1}\left(t-t',\mathbf{x},\mathbf{x}'\right)\right)&=m\left(t-t'\right)\dot{K}^{L+1}\left(t-t',\mathbf{x},\mathbf{x}'\right)K^{d,L}\left(t-t',\mathbf{x},\mathbf{x}'\right)\\ &\quad+e^{-T\sigma^{-2}\left(t-t'\right)}K^{L+1}\left(t-t',\mathbf{x},\mathbf{x}'\right) &(93)\end{aligned}$$

Which is the expression for $K^{d,L+1}\left(t-t'\right)$. Using this identity, we can get a simple expression for the integral over $K^{d,L}\left(t-t'\right)$ at long times:

$$\lim_{t\to\infty}\int_{0}^{t}dt'K^{d,L}\left(t-t',\mathbf{x},\mathbf{x}'\right)=\frac{\sigma^{2}}{T}K_{GP}\left(\mathbf{x},\mathbf{x}'\right) \tag{94}$$

As a result, taking the limit of $t\to\infty$ on both sides of Eq.14, we have

$$\begin{aligned}\lim_{t\to\infty}\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t\right)\right]&=\left(\lim_{t\to\infty}\int_{0}^{t}dt'\mathbf{K}^{d,L}\left(t,t'\right)\right)\left(\boldsymbol{y}-\lim_{t\to\infty}\mathbb{E}\left[\mathbf{f}_{\text{train}}\left(t'\right)\right]\right)\\ \lim_{t\to\infty}\mathbb{E}\left[\mathbf{f}_{\text{train}}(t)\right]&=\mathbf{K}_{GP}^{\top}\left(\mathbf{K}_{GP}+\sigma^{-2}T\boldsymbol{I}\right)^{-1}\boldsymbol{y}\end{aligned} \tag{95}$$

We then take $t\to\infty$ on both sides of Eq.15 and plug in $\lim_{t\to\infty}\mathbb{E}\left[\mathbf{f}_{\text{train}}(t)\right]$ to obtain

$$\begin{aligned}\lim_{t\to\infty}\mathbb{E}\left[\mathrm{f}(\mathbf{x},t)\right]&=\frac{\sigma^{2}}{T}\mathrm{k}_{GP}^{L}(\mathbf{x})^{\top}(\boldsymbol{y}-\lim_{t\to\infty}\mathbb{E}\left[f_{\text{train}}(t)\right])\\ &=\left(\mathbf{k}_{GP}^{L}\left(\mathbf{x}\right)\right)^{\top}\left(\boldsymbol{I}T\sigma^{-2}+\mathbf{K}_{GP}^{L}\right)^{-1}\boldsymbol{y}\end{aligned} \tag{96}$$

which corresponds to Eq.18 in the main text.

C.5 NDK AS A GENERALIZED TWO-TIME NTK

In Eq.7 in the main text, we claimed that the NDK has the following interpretation as a generalized two-time NTK

$$K^{d,L}\left(t,t',\mathbf{x},\mathbf{x}'\right)=e^{-T\sigma^{-2}\left|t-t'\right|}\mathbb{E}\left[\nabla_{\boldsymbol{\theta}_{t}}\mathrm{f}\left(\mathbf{x},\boldsymbol{\theta}_{t}\right)\cdot\nabla_{\boldsymbol{\theta}_{t'}}\mathrm{f}\left(\mathbf{x}',\boldsymbol{\theta}_{t'}\right)\right]_{S_{0}}\,t\geq t' \tag{97}$$

where $\mathbb{E}\left[\cdot\right]_{S_0}$ denotes averaging w.r.t. the prior distribution of the parameters $\boldsymbol{\theta}$, with the statistics defined in Eq.9.

Now we provide formal proof.

We separate $\nabla_{\boldsymbol{\theta}_t}\mathrm{f}\left(\mathbf{x},\boldsymbol{\theta}_t\right)$ into two parts including the derivative w.r.t. the readout weights a_t and the hidden layer weights $\mathcal{W}_t$

Derivative w.r.t. the readout weights:

$$\mathbb{E}\left[\partial_{\mathbf{a}_{t}}\mathrm{f}\left(\mathbf{x},\boldsymbol{\theta}_{t}\right)\cdot\partial_{\mathbf{a}_{t'}}\mathrm{f}\left(\mathbf{x},\boldsymbol{\theta}_{t'}\right)\right]_{S_{0}}=K^{L}\left(t,t',\mathbf{x},\mathbf{x}'\right) \tag{98}$$

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

We have

$$\partial_{\mathbf{W}_t^l} \mathbf{x}_t^L(\mathbf{x}, \mathcal{W}_t) = \frac{1}{\sqrt{N_{L-1} \cdots N_{l-1}}} \Pi_{k=l+1}^{L} \left[\phi'\left(z_t^k\right) \mathbf{W}_t^k\right] \phi'\left(z_t^l\right) \mathbf{x}_t^{l-1} \tag{99}$$

and

$$\begin{aligned}
&\mathbb{E}\left[\partial_{\mathbf{W}_t^l} \mathrm{f}(\mathbf{x}, \boldsymbol{\theta}_t) \cdot \partial_{\mathbf{W}_{t'}^l} \mathrm{f}(\mathbf{x}, \boldsymbol{\theta}_{t'})\right]_{S_0} \\
&= \mathbb{E}\left[N_L^{-1} \mathbf{a}_t \cdot \mathbf{a}_{t'}\right] \left(\Pi_{k=l+1}^{L} \mathbb{E}\left[N_k^{-1} N_{k-1}^{-1} \mathbf{W}_t^k \cdot \mathbf{W}_{t'}^k\right]\right) \left(\Pi_{k=l}^{L} \dot{K}^k(t, t', \mathbf{x}, \mathbf{x}')\right) K^{l-1}(t, t', \mathbf{x}, \mathbf{x}') \\
&= m(t, t')^{L-l+1} \left(\Pi_{k=l}^{L} \dot{K}^k(t, t', \mathbf{x}, \mathbf{x}')\right) K^{l-1}(t, t', \mathbf{x}, \mathbf{x}')
\end{aligned} \tag{100}$$

To leading order in N_l the averages over $\mathbf{a}$ and $\mathcal{W}$ 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.34. The term $m(t, t')$ comes from the covariance of the priors in $\mathcal{W}$ and $\mathbf{a}$, since there are a total of $L - l$ layers of $\mathcal{W}$ and one layer of $\mathbf{a}$, we have $m(t, t')^{L-l+1}$. The kernel $\dot{\mathrm{K}}^k(t, t', \mathbf{x}, \mathbf{x}')$ comes from the inner product between $\phi'\left(z_t^k\right)$ and $\phi'\left(z_{t'}^k\right)$, and the kernel $\mathrm{K}^{l-1}(t, t', \mathbf{x}, \mathbf{x}')$ 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 (Jacot et al. (2018)), we obtain

$$\mathbb{E}\left[\partial_{\mathcal{W}_t} \mathrm{f}(\mathbf{x}, \boldsymbol{\theta}_t) \cdot \partial_{\mathcal{W}_{t'}} \mathrm{f}(\mathbf{x}, \boldsymbol{\theta}_{t'})\right]_{S_0} = e^{T\sigma^{-2}|t-t'|} m(t, t') \dot{K}^L(t, t', \mathbf{x}, \mathbf{x}') K^{d,L-1}(t, t', \mathbf{x}, \mathbf{x}') \tag{101}$$

Combine Eq.101 with Eq.98 and with the definition of $\mathrm{K}^{d,L}(t, t', \mathbf{x}, \mathbf{x}')$ in Eq.67, we have

$$e^{-T\sigma^{-2}|t-t'|} \mathbb{E}\left[\nabla_{\boldsymbol{\theta}_t} \mathrm{f}(\mathbf{x}, \boldsymbol{\theta}_t) \cdot \nabla_{\boldsymbol{\theta}_{t'}} \mathrm{f}(\mathbf{x}', \boldsymbol{\theta}_{t'})\right]_{S_0} = K^{d,L}(t, t', \mathbf{x}, \mathbf{x}') \tag{102}$$

D REPRESENTATIONAL DRIFT

To capture the phenomenon of representational drift, we consider the case where the learning signal stops at some time t_0, while the hidden layers continue to drift according to the dynamics of the prior. If all the weights of the system are allowed to drift, the performance of the mean predictor will deteriorate to chance, thus we consider stable readout weights fixed at the end time of learning t_0. This scenario can be theoretically evaluated using similar techniques to Sec.B.1 , leading to the following equation for the network output:

$$\mathbb{E}\left[f_{\text{drift}}(\mathbf{x}, t, t_0)\right] = \int_0^{t_0} \left(\mathbf{k}^{d,L}(t, t')\right)^\top \left(\boldsymbol{y} - \mathbb{E}\left[\mathbf{f}_{\text{train}}(t')\right]\right) \tag{103}$$

We see here that if $t_0 = t$ it naturally recovers the full mean predictor. It is interesting to look at the limit where the freeze time t_0 is at NNGP equilibrium, where the network has finished its dynamics completely. In this case, the expression can be simplified due to the long-time identity of the NDK (Eq.17 in the main text).

$$\mathbb{E}\left[f_{\text{drift}}(t - t_0)\right] = \left(\mathbf{k}^L(t - t_0)\right)^\top \left(\boldsymbol{I} T\sigma^{-2} + \mathrm{K}_{GP}^L\right)^{-1} \boldsymbol{y} \tag{104}$$

which has a simple meaning of two samples of hidden layer weights from different times at equilibrium. Even at long time differences, the network performance does not decrease to chance, but reaches a new static state.

$$\lim_{t - t_0 \to \infty} \mathbb{E}\left[f_{\text{drift}}(t - t_0)\right] \to \left(\mathbf{k}_{mean}^L(\mathbf{x})\right)^\top \left(\boldsymbol{I} T\sigma^{-2} + \mathrm{K}_{GP}^L\right)^{-1} \boldsymbol{y} \tag{105}$$

We can assess the network's ability to separate classes in a binary classification task by using a linear classifier between the two distributions of outputs (Duda et al. (2000)).

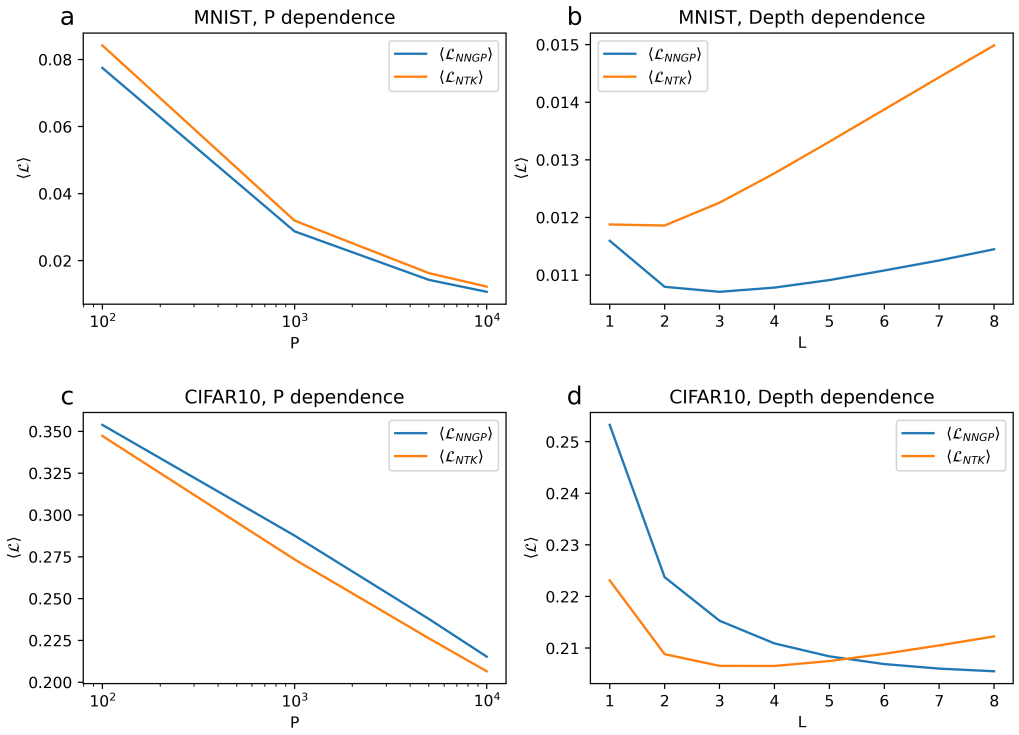

Figure 6: Comparison between NTK and NNGP equilibria, in fully connected DNNs with ReLU activation function.(a-d) The average MSE loss per test example in binary classification tasks of (a,b) MNIST dataset and (c,d) CIFAR10 dataset averaged over all class pairs. We present the results as a function of the number of training examples P (at a constant depth $L = 3$) (a,c), and as a function of depth (at constant $P = 10^4$) (b,d)

D.1 NTK AND NNGP EQUILIBRIA

The NTK and NNGP equilibria mark the initial and final points for the dynamics of the diffusive phase. An interesting question is how different these two equilibria are. In general, the answer depends on the data and the network architecture (Lee et al. (2020)). In Fig.6 we show that in these tasks deeper networks tend to favor the NNGP equilibrium compared with NTK. On the other hand, increasing the size of the training set has a similar effect on both equilibria.

E DETAILS OF THE SIMULATIONS

E.1 SYNTHETIC DATA

We consider P normalized and orthogonal input data vectors $\mathbf{x} \in \mathbb{R}^{N_0}$, such that $\mathbf{K}^{in}_{\mu\nu} = \frac{1}{N_0}\mathbf{x}^\mu \cdot \mathbf{x}^\nu = \delta_{\mu\nu}$. The labels of the data point are ± 1 with equal probability. We consider a test point, which has partial overlap with one of the input vectors, and is orthogonal to all others, w.l.o.g. we assume that the test point is overlapping with the first input vector with label $+1$, such that $\frac{1}{N_0}\mathbf{x}^{test} \cdot \mathbf{x}^\mu = O_{test}\delta_{\mu,1}$, $\frac{1}{N_0}\mathbf{x}^{test} \cdot \mathbf{x}^{test} = 1$, and $y^1 = +1$. In our simulations we set $O_{test} = \frac{3}{4}$, which maximizes the difference between NNGP and NTK equilibria. For this setup, we can represent the kernels by a few scalar functions:

$$\mathbf{K}^{d,L}_{\mu\nu}(t,t') = \mathbf{k}^{d,L}_{\text{offdiag}}(t,t')(1-\delta_{\mu\nu}) + \delta_{\mu\nu}\mathbf{k}^{d,L}_{\text{diag}}(t,t') \tag{106}$$

$$\mathbf{k}^{d,L}_{\mu}(t,t') = \mathbf{k}^{d,L}_{\text{offdiag}}(t,t')(1-\delta_{\mu,1}) + \delta_{\mu,1}\mathbf{k}^{d,L}_{\text{test}}(t,t') \tag{107}$$

Here $\mathbf{k}^{d,L}_{\text{offdiag}}(t,t')$ and $\mathbf{k}^{d,L}_{\text{diag}}(t,t')$ are off-diagonal and diagonal elements of the kernel matrix $\mathbf{K}^{d,L}(t,t')$, they are scalar functions of time, $\mathbf{k}^{d,L}_{\text{test}}(t,t')$ denotes the first element of the vector $\mathbf{k}(t,t')$,and is also a scalar function of both time and the parameter O_{test}. $\mathbf{K}^{L}_{\mu\nu}(t,t')$ and $\dot{\mathbf{K}}^{L}_{\mu\nu}(t,t')$ have the same structure.

Because of the symmetry of this toy model, $\mathbf{f}_{\text{train}}(t)$ takes the same value across all training points with the same label and takes the negative value for training points with the opposite label, and thus can be reduced to a scalar. We consider $\mathbf{f}_{\text{train}}(t)$ on training points with label $+1$. We can transform the vector integral equation into a scalar one, depending only on known scalar functions:

$$\mathbf{f}_{\text{train}}(t) = \int_0^t dt' \left[\left(\mathbf{k}^{d,L}_{\text{diag}}(t,t') - \mathbf{k}^{d,L}_{\text{diag}}(t,t') \right) \left(1 - \mathrm{f}_{train}(t')\right) \right] \tag{108}$$

$$\mathbb{E}\left[\mathrm{f}(\mathbf{x},\boldsymbol{\theta}_t)\right] = \int_0^t dt' \left(\mathbf{k}^{d,L}_{\text{test}}(t,t') - \mathbf{k}^{d,L}_{\text{offdiag}}(t,t') \right) \left(1 - \mathrm{f}_{train}(t')\right) \tag{109}$$

In this model the theoretical results do not depend on P,N_0. For Fig.1, we vary T and σ, σ_0 according to the legend, while keeping $dt = 0.1$.For all other simulations presented, we use $T = 0.001$,$dt = 0.1$, with total time $t = 10000 = 10/T$, $\sigma = 1$, while σ_0 varies depending on $\left(\frac{\sigma_0}{\sigma}\right)^2$ that is presented in the plot.

E.2 MNIST

We consider a digit binary classification task (Deng (2012)), where one type of input is with label $+1$ and the other -1. In our simulations we take digits 1 and 0 as the two classes. We take 50 examples from each class, flatten the image into a vector and normalize the data. The test is a previously unseen example from the class $+1$ to make the comparison with other data sets easy (same with the synthetic data and CIFAR10).The examples in the figures are chosen for a large difference between NTK and NNGP equilibria while the error is relatively small. In Fig.2(e-g) example 25910 from MNIST data set is presented, while in Fig.8 examples 50396 (example 2) and 30508 (example 3) are presented. We used $T = 0.01$,$dt = 0.01$, with total time $t = 1000 = 10/T$. In the simulations presented $\sigma = 1$, while σ_0 varies depending on $\left(\frac{\sigma_0}{\sigma}\right)^2$ that is presented in the plot.

E.3 CIFAR10

We consider an image binary classification task (Krizhevsky et al. (2014)), where one class of input is with label $+1$ and the other -1. In our simulations we take images of cats and dogs as the two classes. We take 50 examples from each class, flatten the image (including channels) into a vector and normalize the data. The test was a previously unseen example from the class $+1$. The examples in the figures are chosen for a large difference between NTK and NNGP equilibria while the error compared to the true label is relatively small. In Fig.2(h) and in Fig.9 example 4484 from CIFAR10 data set is presented (example 1), while in Fig.9 examples 3287 (example 2), 5430 (example 3) and 6433 (example 4) are presented. We used $T = 0.01$,$dt = 0.01$, with total time $t = 1000 = 10/T$, and $\sigma = 1$, σ_0 varies depending on $\left(\frac{\sigma_0}{\sigma}\right)^2$ that is presented in the plot.

E.4 Langevin Dynamics

To check the validity of the theory we performed simulations with Langevin dynamics in a network with $L = 1$, the network is trained under the dynamics given by Eq.26 with $lr = dt = 0.01$, $T = 0.001$, with total time $t = 10000 = 10/T$ on the synthetic data introduced in SI E.1. Simulations shown in Figs.2(a) and Fig.7 are done with $P = 2, N_0 = 100$, and hidden layer width $N = 1000$, $\sigma_0^2/\sigma^2 = 1, 2$ as indicated in the figure captions. Results are averaged over 5000 different initializations and realizations of noise. In the representational drift predictor simulations, at time t_0 the loss changes to contain only the prior part, as presented in Eq.34. The network output was calculated with the hidden layer weights at time t with the readout weights at time t_0.

F ADDITIONAL NUMERICAL RESULTS

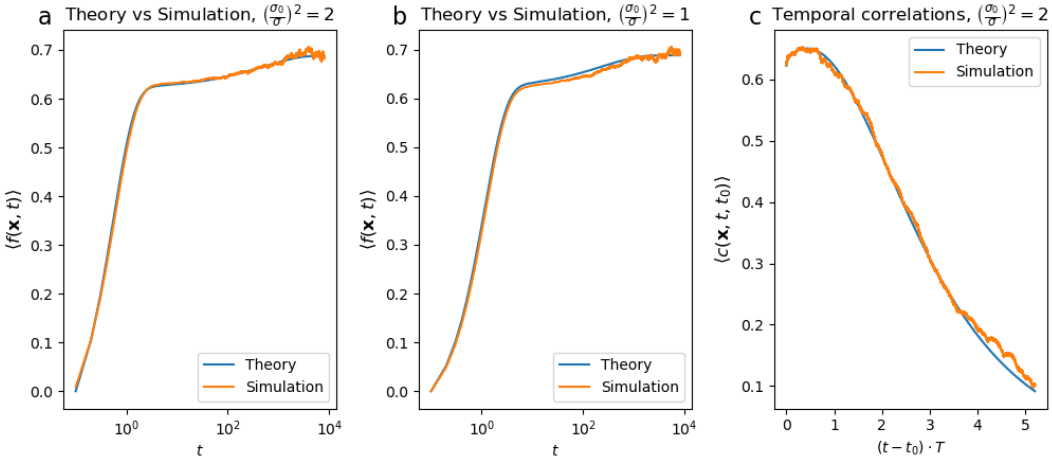

Figure 7: Theory and network simulations of the synthetic data set.(a-b) Theory and simulation of the mean predictor, for different values of $(\sigma_0/\sigma)^2$, with time in log scale due to the large difference in time scales of the two learning phases. (c) Theory and simulation of the temporal correlations of $\mathbf{a}_{t_0}$ at NTK equilibrium with $\mathcal{W}_t$.

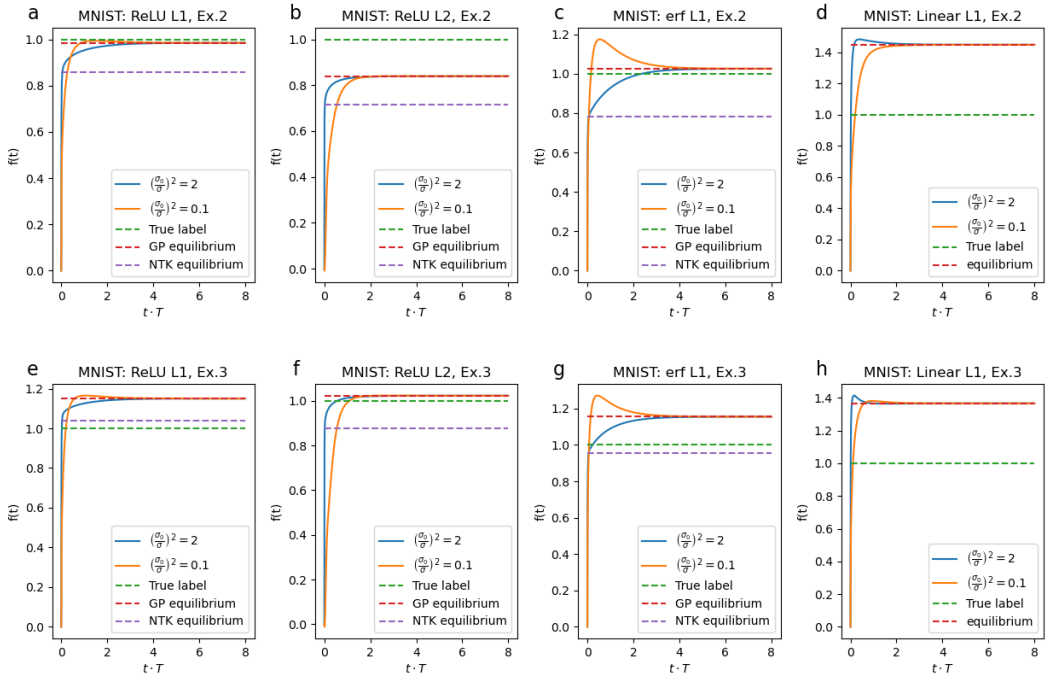

Figure 8: More test examples from MNIST dataset (Deng (2012)), for ReLU ($L = 1, L = 2$), erf and linear activation functions, with different $(\sigma_0/\sigma)^2$ values (a-d) Example 2, with NNGP performance better than NTK. (e-h) Example 3. Interestingly, NTK performance is better than NNGP for ReLU and erf $L = 1$, but is worse for ReLU $L = 2$.

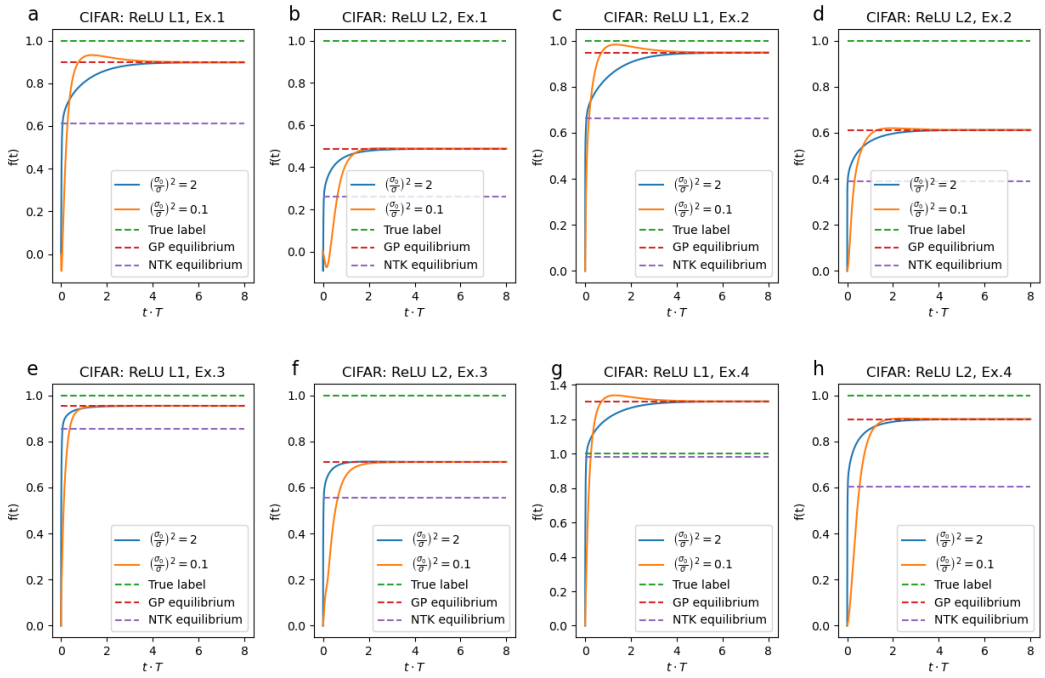

Figure 9: More test examples from CIFAR10 dataset (Krizhevsky et al. (2014)), for ReLU activation function ($L = 1, L = 2$), with different $(\sigma_0/\sigma)^2$ values (a-f) Examples 1,2,3, with NNGP performance better than NTK . (g-h) Example 4. Interestingly, NTK performance is better than NNGP for ReLU $L = 1$, but is worse for ReLU $L = 2$.

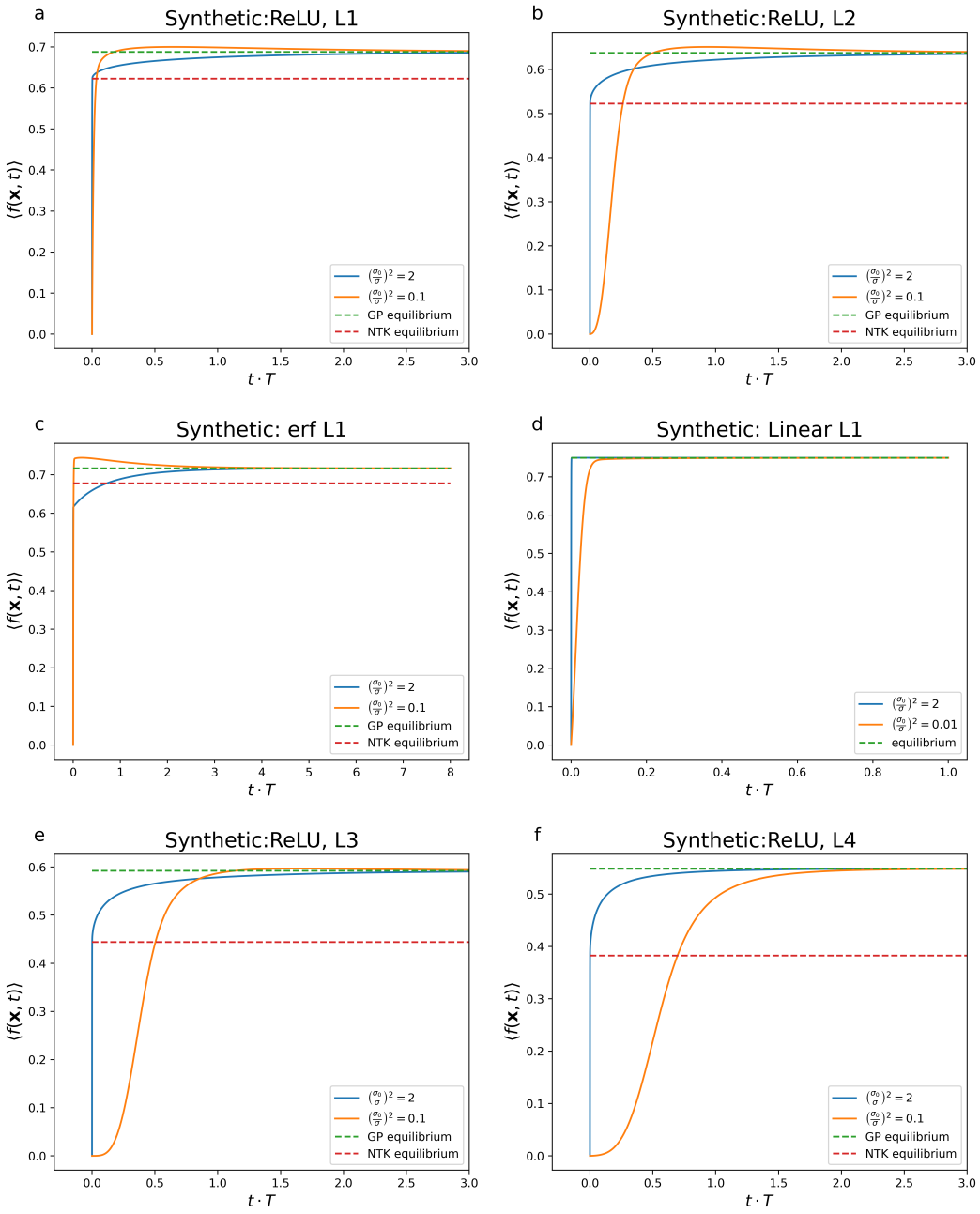

Figure 10: Examples of dynamics for different nonlinearities and depth in the synthetic dataset. (a,b,e,f) ReLU activation function for $L = 1, 2, 3, 4$, with different $(\sigma_0/\sigma)^2$ values. (c) erf activation function with different $(\sigma_0/\sigma)^2$ values. (d) Linear activation function, with different $(\sigma_0/\sigma)^2$ values. We see that with linear activation the system reaches equilibrium in a time shorter than 1/T, during the gradient-driven phase.

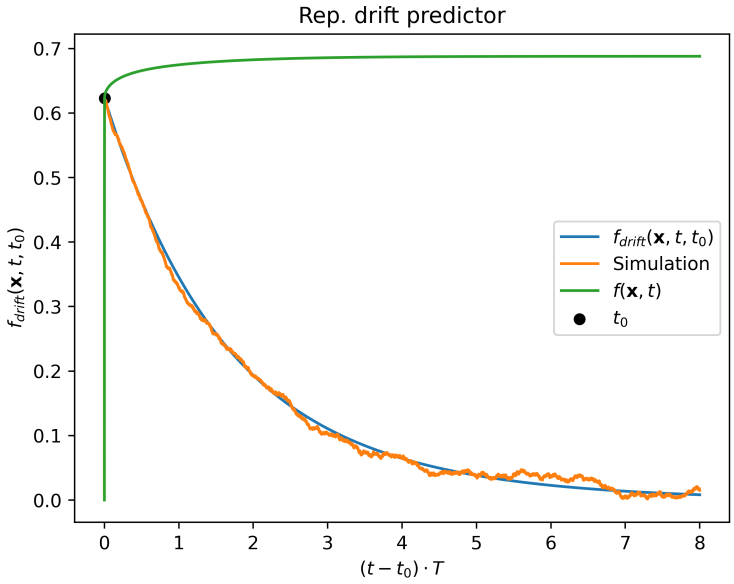

Figure 11: The dynamics of the predictor with no external learning signal and readouts weights frozen at NTK equilibrium (t_0), for the synthetic dataset. We see an approximately exponential decay to chance level performance with time scale of $t \sim 1/T$.