Deep Learning Theory Review: An Optimal Control and Dynamical Systems Perspective

Guan-Horng Liu, Evangelos A. Theodorou

***Abstract*—Attempts from different disciplines to provide a fundamental understanding of deep learning have advanced rapidly in recent years, yet a unified framework remains relatively limited. In this article, we provide one possible way to align existing branches of deep learning theory through the lens of dynamical system and optimal control. By viewing deep neural networks as discrete-time nonlinear dynamical systems, we can analyze how information propagates through layers using mean field theory. When optimization algorithms are further recast as controllers, the ultimate goal of training processes can be formulated as an optimal control problem. In addition, we can reveal convergence and generalization properties by studying the stochastic dynamics of optimization algorithms. This viewpoint features a wide range of theoretical study from information bottleneck to statistical physics. It also provides a principled way for hyper-parameter tuning when optimal control theory is introduced. Our framework fits nicely with supervised learning and can be extended to other learning problems, such as Bayesian learning, adversarial training, and specific forms of meta learning, without efforts. The review aims to shed lights on the importance of dynamics and optimal control when developing deep learning theory.**

***Index Terms*—Deep learning theory, deep neural network, dynamical systems, stochastic optimal control.**

I. Introduction

DEEP learning is one of the most rapidly developing areas in modern artificial intelligence with tremendous impact to different industries ranging from the areas of social media, health and biomedical engineering, robotics, autonomy and aerospace systems. Featured with millions of parameters yet without much hand tuning or domain-specific knowledge, Deep Neural Networks (DNNs) match and sometimes exceed human-level performance in complex problems involving visual synthesizing [1], language reasoning [2], and long-horizon consequential planning [3]. The remarkable practical successes, however, do not come as a free lunch. Algorithms for training DNNs are extremely data-hungry. While insufficient data can readily lead to memorizing irrelevant features [4], data imbalance may cause severe effects such as imposing improper priors implicitly [5]. Although automating tuning for millions of parameters alleviates inductive biases from traditional engineering, it comes with the drawback of making interpretation and analysis difficult. Furthermore, DNN is known for being vulnerable to small adversarial perturbations [6]. In fact, researchers have struggled to improve its robustness beyond infinite norm ball attacks [7]. Finally, while the deep learning community has developed recipes related to the choice of the underlying organization of a DNN, the process of the overall architectural design lacks solid theoretical understanding and remains a fairly ad-hock process.

The aforementioned issues highlight the need towards the development of a theory for deep learning which will provide a scientific methodology to design DNNs architectures, robustify their performance against external attacks/disturbances, and enable the development the corresponding training algorithms. Given this need, our objective in this paper is to review and present in a systematic way work towards the discovery of deep learning theory. This work relies on concepts drawn primarily from the areas of dynamical systems and optimal control theory, and its connections to information theory and statistical physics.

Theoretical understanding of DNN training from previous works has roughly followed two streams: deep latent representation and stochastic optimization. On the topic of deep representations, the composition of affine functions, with element-wise nonlinear activations, plays a crucial role in automatic feature extraction [8] by constructing a chain of differentiable process. An increase in the depth of a NN architecture has the effect of increasing its expressiveness exponentially [9]. This naturally yields a highly over-parametrized model, whose loss landscape is known for a proliferation of local minima and saddle points [10]. However, the over-fitting phenomena, suggested by the bias-variance analysis, has not been observed during DNN training [11]. In practice, DNN often generalizes remarkably well on unseen data when initialized properly [12].

Generalization of highly over-parametrized models cannot be properly explained without considering stochastic optimization algorithms. Training DNN is a non-convex optimization problem. Due to its high dimensionality, most practically-used algorithms utilize first-order derivatives with aids of adaptation and momentum mechanisms. In fact, even a true gradient can be too expensive to compute on the fly; therefore only an unbiased estimation is applied at each update. Despite these approximations that are typically used to enable applicability, first-order stochastic optimization is surprisingly robust and algorithmically stable [13]. The stochasticity stemmed from estimating gradients is widely believed to perform implicit regularization [14], guiding parameters towards flat plateaus with lower generalization errors [15]. First-order methods are also proven more efficient to escape from saddle points [16], whose

The authors are with the Autonomous Control and Decision Systems (ACDS) Lab, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA (e-mail: ghliu@gatech.edu; evangelos.theodorou@gatech.edu)

number grows exponentially with model dimensionality [10]. Research along this line provides a fundamental understanding on training dynamics and convergence property, despite the analysis is seldom connected to the deep representation viewpoint.

How do the two threads of deep latent organization and stochastic optimization interplay with each other and what are the underlying theoretical connections? These are questions that have not been well-explored and are essential towards the development of a theory for Deep Learning. Indeed, study of stochastic optimization dynamics often treats DNN merely as a black-box. This may be insufficient to describe the whole picture. When using back-propagation [17] to obtain first-order derivatives, the backward dynamics, characterized by the compositional structure, rescales the propagation made by optimization algorithms, which in return leads to different representation at each layer. Frameworks that are able to mathematically characterize these compounding effects will provide more nuanced statements. One of such attempts has been information bottleneck theory [18], which describes the dynamics of stochastic optimization using information theory, and connects it to optimal representation via the bottleneck principle. Another promising branch from Du *et al.* [19], [20] showed that the specific representation, i.e. the Gram matrix, incurred from gradient descent (GD) characterizes the dynamics of the prediction space and can be used to prove global optimality. These arguments, however, have been limited to either certain architectures [21] or noise-free optimization[1].

In this review, we provide a dynamical systems and optimal control perceptive to DNNs in an effort to systematize the alternative approaches and methodologies. This allows us to pose and answer the following questions: (i) at which state should the training trajectory start, i.e. how should we initialize the weights or hyper-parameters, (ii) through which path, in a distribution sense, may the trajectory traverse, i.e. can we give any prediction of training dynamics on average, (iii) to which fixed point, if exists, may the trajectory converge, and finally (iv) the stability and generalization property at that fixed point. In the context of deep learning, these can be done by recasting DNNs and optimization algorithms as (stochastic) dynamical systems. Advanced tools from signal processing, mean-field theory, and stochastic calculus can then be applied to better reveal the training properties. We can also formulate an optimal control problem upon the derived dynamics to provide principled guidance for architecture and algorithm design. The dynamical and control viewpoints fit naturally with supervised learning and can readily extend to other learning schemes, such as Bayesian learning, adversarial training, and specific forms of meta learning, This highlights the potential to provide more theoretical insights.

The article is organized as follows. In Sec. II, we will go over recent works related to the dynamical viewpoint. Sec. III and IV demonstrate how we can recast DNNs and stochastic optimizations as dynamical systems, then apply control theory to them. In Sec. V, we extend our analysis to other learning problems. Finally, we conclude our work and discuss some future directions in Sec. VI.

Notation: We will denote $\boldsymbol{h}_l$ and $\boldsymbol{x}_l$ as the (pre-)activation at layer l ($\boldsymbol{x}_0 \equiv \boldsymbol{x}$ for simplicity). Mapping to the next layer obeys $\boldsymbol{h}_l = \mathcal{W}(\boldsymbol{x}_l; \boldsymbol{\theta}_l)$ and $\boldsymbol{x}_{l+1} = \phi(\boldsymbol{h}_l)$, where ϕ is a nonlinear activation function and $\mathcal{W}$ is an affine transform parametrized by $\boldsymbol{\theta}_l \in \mathbb{R}^m$. The full parameter space across all layers is denoted $\boldsymbol{\theta} \equiv \{\boldsymbol{\theta}_l\}_{l=0}^{L-1} \in \mathbb{R}^{\bar{m}}$. Given a data set $\mathcal{D} := \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_i$, where $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^d$, we aim to minimize a loss $\Phi(\cdot)$, or equivalently the cost $J(\cdot)$ from the control viewpoint. The element of the vector/matrix are respectively denoted as $\mathbf{a}^{(i)} \equiv \mathrm{a}_i$ and $\mathbf{A}^{(i,j)} \equiv \mathrm{A}_{(i,j)}$. We will follow the convention $\langle \cdot, \cdot \rangle$ to denote the inner product of two vectors, with $\langle f(\cdot), g(\cdot) \rangle_\mu := \int_\Omega f(w)^\mathsf{T} g(w) \mathrm{d}\mu(w)$ as its generalization to the inner product of two functions weighted by a probability measure. We will always use the subscript t to denote the dynamics. Depending on the context, it can either mean propagation through DNN (Sec. III) or training iterations (Sec. IV).

II. Related Work

A. Mean Field Approximation & Gaussian Process

Mean field theory allows us to describe distributions of activations and pre-activations over an ensemble of untrained DNNs in an analytic form. The connection was adapted in [9] to study how signals propagate through layers at the initialization stage. It implies an existence of an *order-to-chaos* transition as a function of parameter statistics. While networks in the phase of *order* suffer from saturated information and vanished gradients, in the *chaotic* regime expressions of networks grow exponentially with depth, and exploded gradients can be observed. This phase transition diagram, formally characterized in [24], provides a necessary condition towards network *trainability* and determines an upper bound on the number of layers allowable for information to pass through. This analysis has been successfully applied to most commonly-used architectures for critical initialization, including FCN, CNN, RNN, LSTM, ResNet [24]–[28]. In addition, it can also be used to provide geometric interpretation by estimating statistical properties of Fisher information [29], [30]. It is worth noticing that all aforementioned works require the limit of layer width and i.i.d. weight priors in order to utilize the Gaussian approximation. Indeed, the equivalence between DNN and Gaussian process has long been known for single-layer neural networks [31] and extended to deeper architectures recently [32], [33]. The resulting Bayesian viewpoint enables uncertainty estimation and accuracy prediction at test time [34].

B. Implicit Bias in Stochastic Gradient Descent (SGD)

There has been commensurate interest in studying the implicit bias stemmed from stochastic optimization. Even without stochasticity, vanilla GD algorithms are implicitly regulated as they converge to max-margin solutions for both linear

[1] We note that global optimality for stochastic gradient descent has been recently proven in [22], [23], yet their convergence theories rely on certain assumptions on the data set in order to have the Frobenius norm of the (stochastic) gradient lower-bounded. This is in contract the least eigenvalue of the prediction dynamics in [19], [20], which is more related to the dynamical analysis in this review.

predictors [35], [36] and over-parametrized models, e.g. multi-layers linear networks [37], [38] and shallow neural networks with nonlinear activations [39]. When the stochasticity is introduced, a different regularization effect has been observed [40]. This implicit regularization plays a key role in explaining why DNNs generalize well despite being over-parametrized [11], [41]. Essentially, stochasticity pushes the training dynamics away from sharp local minima [42] and instead guides it towards flat plateaus with lower generalization errors [15]. An alternative view suggests a convolved, i.e. smoothening, effect on the loss function throughout training [43]. The recent work from Chaudhari *et al.* [14] provides mathematical intuitions by showing that SGD performs variational inference under certain approximations. Algorithmically, Chaudhari *et al.* proposed a surrogate loss that explicitly biases SGD dynamics towards flat local minima [44], [45]. The corresponding algorithm relates closely to stochastic gradient Langevin dynamics, a computationally efficient sampling technique originated from Markov Chain Monte Carlo (MCMC) for large-scale problems [46], [47].

C. Information Theory & Statistical Physics

Research along this direction studies the dynamics of Markovian stochastic process and its effect on the deep representation at an ensemble level. For instance, the Information Bottleneck (IB) theory [18], [48] studies the training dynamics on the Information Plane described by the mutual information of layer-wise activations. The principle of information bottleneck mathematically characterizes the phase transition from memorization to generalization, mimicking the critical learning periods in biological neural systems [49]. Applying the same information Lagrangian to DNN's weights has revealed intriguing properties of the deep representation, such as invariance, disentanglement and generalization [50], [51], despite a recent debate in [21] arguing the inability of the findings in references [18], [48] to generalize beyond certain architectures. In [52], similar statements on the implicit bias has been drawn from the Algorithmic Information Theory (AIT), suggesting the parameter-function map of DNNs is exponentially biased towards simple functions. The information theoretic viewpoint is closely related to statistical physics. In [53], [54], an upper bound mimicking the second law of stochastic thermodynamics was derived for single-layer networks on binary classification tasks. In short, generalization of a network to unseen datum is bounded by the summation of the Shannon entropy of its weights and a term that captures the total heat dissipated during training. The concept of *learning efficiency* was introduced as an alternative metric to compare algorithmic performance. Additionally, Yaida *et al.* [55] derived a discrete-time master equation at stationary equilibrium and linked it to the fluctuation-dissipation theorem in statistical mechanics.

D. Dynamics and Optimal Control Theory

The dynamical perspective has received considerable attention recently as it brings new insights to deep architectures and training processes. For instance, viewing DNNs as a discretization of continuous-time dynamical systems is proposed in [56]. From such, the propagating rule in the deep residual network [57], $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + f(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$, can be thought of as an one-step discretization of the forward Euler scheme on an ordinary differential equation (ODE), $\dot{\boldsymbol{x}}_t = f(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$. This interpretation has been leveraged to improve residual blocks in the sense that it achieves more effective numerical approximation [58]. In the continuum limit of depth, the flow representation of DNNs has made the transport analysis with Wasserstein geometry possible [59]. Algorithmically, efficient computational methods have been developed in [60], [61] to allow parameterization of (stochastic) continuous-time dynamics (e.g. derivative of latent variables) directly with DNNs. When the analogy between optimization algorithms and controllers is further drawn [62], standard supervised learning can be recast as a mean-field optimal control problem [63]. This is particularly beneficial since it enables new training algorithms inspired from optimal control literature [64]–[66].

Similar analysis can be applied to SGD by viewing it as a stochastic dynamical system. In fact, most previous discussions on implicit bias formulate SGD as stochastic Langevin dynamics [67]. Other stochastic modeling, such as Lèvy process, has been recently proposed [68]. In parallel, stability analysis of the Gram matrix dynamics induced by DNN reveals global optimality of GD algorithms [19], [20]. Applying optimal control theory to SGD dynamics results in optimal adaptive strategies for tuning hyper-parameters, such as the learning rate, momentum, and batch size [69], [70].

III. Deep Neural Network as a Dynamical System

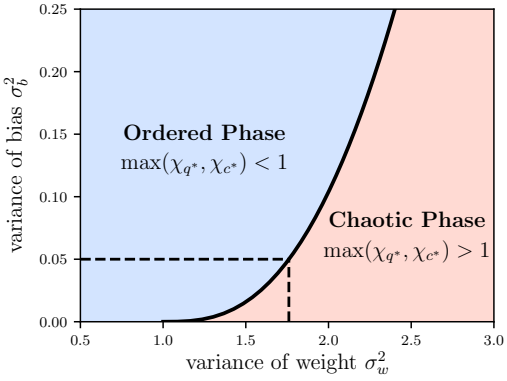
As mentioned in Sec. II, DNNs can be interpreted as finite-horizon nonlinear dynamical systems by viewing each layer as a distinct time step. In Sec III-A, we will discuss how to explore this connection to analyze information propagation inside DNN. The formalism establishes the foundation of recent works [19], [24], [27], and we will discuss its implications in Sec. III-B. In Sec III-C, we will draw the connection between optimization algorithms and controllers, leading to an optimal control formulation of DNN training characterized by mean-field theory. Hereafter we will focus on fully-connected (FC) layers and leave remarks for other architectures, e.g. convolution layers and residual connections, in Appendix A.

A. Information Propagation inside DNN

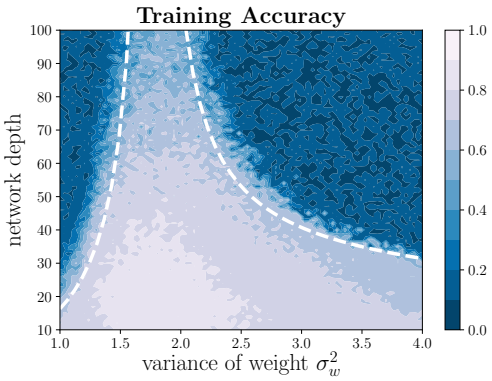
Recall the dynamics of a FC-DNN at time step t, i.e. at layer t-th and suppose its weights and biases are initialized by drawing i.i.d. from two zero-mean Gaussians, i.e.

$$\begin{aligned} \mathbf{h}_t = \mathcal{W}_{\text{FC}}(\mathbf{x}_t; \boldsymbol{\theta}_t) := \mathbf{W}_t \mathbf{x}_t + \mathbf{b}_t, \text{ where} \\ \mathbf{W}_t^{(i,j)} \sim \mathcal{N}(0, \sigma_w^2/N_t) \quad \mathbf{b}_t^{(i)} \sim \mathcal{N}(0, \sigma_b^2) \ . \end{aligned} \tag{1}$$

We denote $\boldsymbol{\theta}_t \equiv (\mathbf{W}_t, \mathbf{b}_t)$. σ_w^2 and σ_b^2 respectively represent the variance of the weights and biases, and N_t is the dimension of the pre-activation at time t. Central limit theorem (CLT) implies in the limit of large layer widths, $N_t \gg 1$, the distribution of pre-activation elements, $\mathbf{h}_t^{(i)}$, also converges to



(a) The phase diagram of $\max(\chi_{q^*}, \chi_{c^*})$ as the function of σ_w and σ_b. The solid line represents the critical initialization where information inside DNN is neither saturated, as in the ordered phase, nor exploding, as in the chaotic phase.



(b) Prediction of the network trainability given its depth and σ_w^2, with σ_b^2 fixed to 0.05. The color bar represents the training accuracy on MINST after 200 training steps using SGD. It is obvious that the boundary at which networks become untrainable aligns with the theoretical depth scale, denoted white dashed line, up to a constant ($\sim 4.5\xi_{c^*}$ in this case). Also, notice that the peak around $\sigma_w^2 = 1.75$ is precisely predicted by the critical line in Fig. 1a for $\sigma_b^2 = 0.05$.

Fig. 1. Reproduced results[2] from [24] for random FC-DNNs with $\phi = \tanh$.

a Gaussian. It is straightforward to see the distribution also has zero mean and its variance can be estimated by matching the second moment of the empirical distribution of $\mathbf{h}_t^{(i)}$ across all N_t,

$$q_t := \frac{1}{N_t}\sum_{i=0}^{N_t}(\mathbf{h}_t^{(i)})^2 \ . \tag{2}$$

q_t can be viewed as the normalized squared length of the pre-activation, and we will use it as the statistical quantity of the information signal. The dynamics of q_t, when propagating from time step t to $t+1$, takes a nonlinear form

$$q_{t+1} = \sigma_w^2 \mathbb{E}_{\mathbf{h}_t^{(i)}\sim\mathcal{N}(0,q_t)}\left[\phi^2(\mathbf{h}_t^{(i)})\right] + \sigma_b^2 \ , \tag{3}$$

with the initial condition given by $q_0 = \frac{1}{N_0}\boldsymbol{x}_0\cdot\boldsymbol{x}_0$. Notice that despite starting the derivation from random neural networks, the mapping in (3) admits a deterministic process, depending only on σ_w, σ_b, and $\phi(\cdot)$. We highlight this determinism as the benefit gained by mean-field approximation. Schoenholz *et al.* [24] showed that for any bounded ϕ and finite value of σ_w and σ_b, there exists a fixed point, $q^* := \lim_{t\to\infty} q_t$, regardless of the initial state q_0.

Similarly, for a pair of input $(\boldsymbol{x}^{(\alpha)}, \boldsymbol{x}^{(\beta)})$ we can derive the following recurrence relation

$$q_{t+1}^{(\alpha,\beta)} = \sigma_w^2 \mathbb{E}_{(\mathbf{h}_t^{(\alpha)},\mathbf{h}_t^{(\beta)})^\intercal\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_t)}[\phi(\mathbf{h}_t^{(\alpha)})\phi(\mathbf{h}_t^{(\beta)})] + \sigma_b^2 \tag{4}$$

$$\text{where} \quad \boldsymbol{\Sigma}_t = \begin{pmatrix} q_t^{\alpha} & q_t^{(\alpha,\beta)} \\ q_t^{(\alpha,\beta)} & q_t^{\beta} \end{pmatrix} \tag{5}$$

is the covariance matrix at time t and the initial condition is given by $q_0^{(\alpha,\beta)} = \frac{1}{N_0}\boldsymbol{x}_0^{(\alpha)}\cdot\boldsymbol{x}_0^{(\beta)}$. Under the same conditions for q^* to exist, we also have the fixed points for the covariance and correlation, respectively denoted $q^*_{(\alpha,\beta)}$ and $c^* = q^*_{(\alpha,\beta)}/\sqrt{q^*_\alpha q^*_\beta} = q^*_{(\alpha,\beta)}/q^*$. The element of the covariance matrix at its fixed point $\boldsymbol{\Sigma}^*$ hence takes a compact form

$$\Sigma^*_{(\alpha,\beta)} = q^*\left(\delta_{(\alpha,\beta)} + (1-\delta_{(\alpha,\beta)})c^*\right) \ , \tag{6}$$

where $\delta_{(a,b)}$ is Kronecker delta. It is easy to verify that $\boldsymbol{\Sigma}^*$ has q^* for the diagonal entries and q^*c^* for the off-diagonal ones. In short, when the mean field theory is applied to approximate distributions of activations and pre-activations, the statistics of the distribution follows a deterministic dynamic as propagating through layers. This statistic can be treated as the information signal and from there the dynamical system analysis can be applied, as we will show in the next subsection.

B. Stability Analysis & Implications

Traditional stability analysis of dynamical systems often involves computing the Jacobian at the fixed points. The Jacobian matrix describes the rate of change of system output when small disturbance is injected to input. In this vein, we can define the residual system, $\boldsymbol{\epsilon}_t := \boldsymbol{\Sigma}_t - \boldsymbol{\Sigma}^*$, and first-order expand[3] it at $\boldsymbol{\Sigma}^*$. The independent evolutions of the two signal quantities, as shown in (3) and (4), already hint that the Jacobian can be decoupled into two sub-systems. Their eigenvalues are given by,

$$\chi_{q^*} = \sigma_w^2 \mathbb{E}_{\mathbf{h}\sim\mathcal{N}(0,\boldsymbol{\Sigma}^*)}[\phi''(\mathbf{h}^{(i)})\phi(\mathbf{h}^{(i)}) + \phi'(\mathbf{h}^{(i)})^2] \tag{7}$$

$$\chi_{c^*} = \sigma_w^2 \mathbb{E}_{\mathbf{h}\sim\mathcal{N}(0,\boldsymbol{\Sigma}^*)}[\phi'(\mathbf{h}^{(i)})\phi'(\mathbf{h}^{(j)})] \quad \mathbf{h}^{(i)} \neq \mathbf{h}^{(j)} \ . \tag{8}$$

This eigen-decomposition suggests the information traverses through layers in the diagonal and off-diagonal eigenspace. A fixed point is stable if and only if both χ_{q^*} and χ_{c^*} are less than 1. In fact, the logarithms of χ_{q^*} and χ_{c^*} relate to

[2]Code is available in https://github.com/ghliu/mean-field-fcdnn.

[3]We refer readers to the Supplementary Material in [24] for a complete treatment.

the well-known Lyapunov exponents in the dynamical system theory, i.e.

$$|q_t - q^*| \sim e^{-t/\xi_{q^*}} \qquad \xi_{q^*}^{-1} = -\log \chi_{q^*} \quad \text{and} \tag{9}$$

$$|c_t - c^*| \sim e^{-t/\xi_{c^*}} \qquad \xi_{c^*}^{-1} = -\log \chi_{c^*} \quad , \tag{10}$$

where c_t denotes the dynamics of the correlation.

The dynamical system analysis in (7-10) has several important implications. Recall again that given a DNN, its propagation rule depends only on σ_w and σ_b. We can therefore construct a phase diagram with σ_w and σ_b as axes and define a critical line that separates the ordered phase, in which all eigenvalues are less than 1 to stabilize fixed points, and the chaotic phase, in which either χ_{q^*} or χ_{c^*} exceeds 1, leading to divergence. An example of the phase diagram for FC-DNNs is shown in Fig. 1a. Networks initialized in the ordered phase may suffer from saturated information if the depth is sufficiently large. They become un-trainable since neither forward nor backward propagation is able to penetrate to the destination layer. Fig. 1b gives an illustration of how ξ_{q^*} and ξ_{c^*}, named *depth scale* [24], predict the trainability of random DNNs. On the other hand, networks initialized along the critical line remain marginal stable, and information is able to propagate through an arbitrary depth without saturating or exploding. It is particularly interesting to note that while the traditional study on dynamical systems focuses on stability conditions, the dynamics inside DNN instead requires a form of transient chaos.

The discussion of the aforementioned critical initialization, despite being crucial in the success of DNN training [71], may seem limited since once the training process begins, the i.i.d. assumptions in order to construct (2), and all those derivations afterward, no longer hold. Fortunately, it has been empirically observed that when DNN is sufficiently over-parameterized, its weight will be close to the random initialization over training iterations [72]. In other words, under certain assumptions, the statistical property derived at initialization can be preserved throughout training. This has a strong implication as it can be leveraged to prove the global convergence and optimality of GD [19], [20]. Below we provide the proof sketch and demonstrate how the deep information is brought into the analysis.

Recalling the information defined in (4-5) for a pair of input, we can thereby construct a *Gram matrix* $\mathbf{K}_t \in \mathbb{R}^{|\mathcal{D}|\times|\mathcal{D}|}$, where $|\mathcal{D}|$ is the size of the dataset and $t \in \{0, 1, \cdots, T-1\}$. The element of $\mathbf{K}_t$ represents the information quantity between the data points with the corresponding indices, i.e. $\mathbf{K}_t^{(i,j)} := q_t^{(i,j)}$. The Gram matrix can be viewed as a deep representation of a matrix form induced by the DNN architecture and dataset at random initialization. The same matrix has been derived in several concurrent works, namely the Neural Tangent Kernel (NTK) in [73].

Now, consider a mean squared loss, $\frac{1}{2}\|\mathbf{y} - \mathbf{u}\|_2^2$, where $\mathbf{y}, \mathbf{u} \in \mathbb{R}^{|\mathcal{D}|}$ and each element $\mathbf{u}^{(i)} := \mathbf{1}^\intercal \boldsymbol{x}_T^{(i)} / \|\mathbf{1}\|_2$ denotes the scalar prediction of each data point $i \in \mathcal{D}$. The dynamics of the prediction error governed by the GD algorithm takes an analytical form [20] written as

$$\begin{aligned} \mathbf{y} - \mathbf{u}(k+1) \approx & (\mathbf{I} - \eta \mathbf{G}(k))(\mathbf{y} - \mathbf{u}(k)) \ , \\ \text{where } \mathbf{G}^{(i,j)}(k) := & \langle \frac{\partial \mathrm{u}_i(k)}{\partial \theta_{T-1}(k)}, \frac{\partial \mathrm{u}_j(k)}{\partial \theta_{T-1}(k)} \rangle \ . \end{aligned} \tag{11}$$

k and η denote the iteration process and learning rate. Equation (11) indicates a linear dynamics characterized by the matrix $\mathbf{G}$, whose element at initialization is related to the one of $\mathbf{K}$ by[4]

$$\begin{aligned} \mathbf{G}^{(i,j)}(0) &= \frac{1}{\sigma_w^2} \mathbf{K}_{T-1}^{(i,j)} \mathbb{E}_{\mathbf{h}\sim\mathcal{N}(0,\boldsymbol{\Sigma}_{T-1})}[\phi'(\mathbf{h}^{(i)})\phi'(\mathbf{h}^{(j)})] \\ &=: \mathbf{K}_T^{(i,j)} \ . \end{aligned} \tag{12}$$

When the width is sufficiently large, $\mathbf{G}(k)$ will be close to $\mathbf{K}_T$ (precisely $\|\mathbf{G}(k) - \mathbf{K}_T\|_2$ is bounded) for all iterations $k \geq 0$. This, together with the least eigenvalue of $\mathbf{K}_T$ being lower-bounded for non-degenerate dataset, concludes the linear convergence to the global minimum.

C. Training DNN with Optimal Control

To frame optimization algorithms and training processes into the dynamical system viewpoint, one intuitive way is to interpret optimization algorithms as controllers. As we will show in Sec. III-C1, this can be achieved without loss of generality and naturally yields an optimal control formalism of the deep learning training process. Such a connection is useful since the optimality conditions of the former problem are well studied and characterized by the Pontryagin's Minimum Principle (PMP) and the Hamilton-Jacobi-Bellman (HJB) equation, which we will introduce in Sec. III-C2 and III-C3, respectively. The fact that back-propagation [17] can be viewed as an approximation of PMP [64] opens a room for new optimization algorithms inspired from the optimal control perspective.

1) Mean-Field Optimal Control Derivation: In [62], a concrete connection was derived between first-order optimization algorithms and PID controllers. To see that, consider the formula of gradient descent and the discrete realization of integral control:

$$\theta_{k+1} = \theta_k - \nabla f(\theta_k) \tag{13}$$

$$u_{k+1} = \sum\nolimits_{i=0}^{k} e_i \cdot \Delta t \tag{14}$$

These two update rules are equivalent when we interpret the gradient $-\nabla f(\theta_k)$ as tracking error $e(k)$. Both modules are designed to drive certain statistics of a system, either the loss gradient or tracking error, towards zero, by iteratively updating new variables to affect system dynamics. When a momentum term is introduced, it will result in an additional lag component which helps increase the low-frequency loop gain [62]. This suggests accelerated convergence near local minima, as observed in the previous analysis [74]. In other words, the parameters in DNNs can be recast as the control variables in dynamical systems.

[4] we set σ_b^2 to 0 in (12) to match the formulation used in [19], [20].

With this viewpoint in mind, we can draw an interesting connection between deep learning training processes and optimal control problems (15). In a vanilla discrete-time OCP, the minimization problem takes the form

$$\min_{\{\boldsymbol{\theta}_t\}_{t=0}^{T-1}} J := \left[\Phi(\boldsymbol{x}_T) + \sum_{t=0}^{T-1} L(\boldsymbol{x}_t, \boldsymbol{\theta}_t)\right] \quad \text{s.t. } \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{\theta}_t) \ , \tag{15}$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{\theta}_t \in \mathbb{R}^m$ represent state and control vectors. f, L and Φ respectively denote the dynamics, intermediate cost and terminal cost functions. In this vein, the goal of (supervised) learning is to find a set of optimal parameters at each time step (i.e. layer), $\{\boldsymbol{\theta}_t\}_{t=0}^{T-1}$, such that when starting from the initial state $\boldsymbol{x}_0$, its terminal state $\boldsymbol{x}_T$ is close to the target $\boldsymbol{y}$. Dynamical constraints in (15) are characterized by the DNNs, whereas terminal and control-dependent intermediate costs correspond to training loss and weight regularization. Though state-dependent intermediate costs are not commonly-seen in supervised problems until recently [75], it has been used extensively in the context of deep reinforcement learning to guide or stabilize training, e.g. the auxiliary tasks and losses [76], [77].

Extending (15) to accept batch data requires viewing the input-output pair $(\boldsymbol{x}_0, \boldsymbol{y})$ as a random variable drawn from a probability measure. This can be done by introducing the mean-field formalism where the analysis is lifted to distribution spaces. The problem becomes searching an optimal transform that propagates the input population to the desired target distribution. The population risk minimization problem can hence be regarded as a mean-field optimal control problem [63],

$$\inf_{\boldsymbol{\theta}_t \in L^\infty} \mathbb{E}_{(\mathbf{x}_0, \mathbf{y}) \sim \mu_0} \left[\Phi\left(\mathbf{x}_T, \mathbf{y}\right) + \int_0^T L\left(\mathbf{x}_t, \boldsymbol{\theta}_t\right) dt\right] \quad \text{s.t. } \dot{\mathbf{x}}_t = f(\mathbf{x}_t, \boldsymbol{\theta}_t) \ , \tag{16}$$

where $L^\infty \equiv L^\infty([0,T], \mathbb{R}^m)$ denotes the set of essentially-bounded measurable functions and μ_0 is the joint distribution of the initial states $\mathbf{x}_0$ and terminal target $\mathbf{y}$. Note that we change our formulation from the discrete-time realization to the continuous-time framework since it is mathematically easier to analyze and offers more flexibilities. The formulation (16) allows us to analyze the optimization of DNN training through two perspectives, namely the minimum principle approach and dynamic programming approach, as we will now proceed.

2) Mean-Field Pontryagin's Minimum Principle: The necessary conditions of the problem (15) are described in the celebrated Pontryagin's Minimum Principle (PMP) [78]. It characterizes the conditions that an optimal state-control trajectory must obey locally. E *et al.* [63] derived the mean-field extension of the theorem, which we will restate below. We will focus on its relation with standard DNN optimization, i.e. gradient descent with back-propagation, and refer to [63] for the concrete proof.

Theorem 1 (Mean-Field PMP [63])**.** *Assume the following statements are true:*

(A1) The function f is bounded; f, L are continuous in $\boldsymbol{\theta}_t$; and f, L, Φ are continuously differentiable with respect to $\mathbf{x}_t$.

(A2) The distribution μ_0 has bounded support, i.e. $\mu_0\left(\left\{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^n \times \mathbb{R}^d : \|\boldsymbol{x}\| + \|\boldsymbol{y}\| \le M\right\}\right) = 1$ for some $M > 0$.

Let $\boldsymbol{\theta}_t^* : t \mapsto \mathbb{R}^m$ be a solution that achieves the infimum of (16). Then, there exists continuous stochastic processes $\mathbf{x}_t^*$ and $\mathbf{p}_t^*$, such that

$$\dot{\mathbf{x}}_t^* = \nabla_{\mathbf{p}} H\left(\mathbf{x}_t^*, \mathbf{p}_t^*, \boldsymbol{\theta}_t^*\right), \qquad \mathbf{x}_0^* = \mathbf{x}_0 \ , \tag{17}$$

$$\dot{\mathbf{p}}_t^* = -\nabla_{\mathbf{x}} H\left(\mathbf{x}_t^*, \mathbf{p}_t^*, \boldsymbol{\theta}_t^*\right), \quad \mathbf{p}_T^* = \nabla_{\mathbf{x}} \Phi\left(\mathbf{x}_T^*, \mathbf{y}\right) \ , \tag{18}$$

$$\forall \boldsymbol{\theta}_t \in \mathbb{R}^m, \quad a.\ e.\ t \in [0,T] \ , \quad \mathbb{E}_{\mu_0} H\left(\mathbf{x}_t^*, \mathbf{p}_t^*, \boldsymbol{\theta}_t^*\right) \le \mathbb{E}_{\mu_0} H\left(\mathbf{x}_t^*, \mathbf{p}_t^*, \boldsymbol{\theta}_t\right) \ , \tag{19}$$

where the Hamiltonian function $H : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is given by

$$H\left(\boldsymbol{x}_t, \boldsymbol{p}_t, \boldsymbol{\theta}_t\right) = \boldsymbol{p}_t \cdot f(\boldsymbol{x}_t, \boldsymbol{\theta}_t) + L(\boldsymbol{x}_t, \boldsymbol{\theta}_t) \tag{20}$$

and $\boldsymbol{p}_t$ denotes the co-state of the adjoint equation.

Theorem 1 resembles the classical PMP result except that the Hamiltonian minimization condition (19) is now taken over an expectation w.r.t. μ_0. Also, notice the optimal control trajectory $\boldsymbol{\theta}_t^*$ admits an open-loop process in the sense that it does not depend on the population distribution. This is in contrast to what we will see from the dynamic programming approach (i.e. Theorem 2).

The conditions characterized by (17-19) can be linked to the optimization dynamics in DNN training. First, (17) is simply the feed-forward pass from the first layer to the last one. The co-state can be interpreted as the Lagrange multiplier of the objective function w.r.t. the constraint variables [79], and its backward dynamics is described in (18). Here, we shall regard (18) as the back-propagation [64]. To see that, consider the discrete-time Hamiltonian function,

$$H\left(\boldsymbol{x}_t, \boldsymbol{p}_{t+1}, \boldsymbol{\theta}_t\right) = \boldsymbol{p}_{t+1} \cdot f(\boldsymbol{x}_t, \boldsymbol{\theta}_t) + L(\boldsymbol{x}_t, \boldsymbol{\theta}_t) \ . \tag{21}$$

Substituting it into the discrete-time version of (18) will lead to the chain rule used derive back-propagation,

$$\begin{aligned} \boldsymbol{p}_t^* &= \nabla_{\boldsymbol{x}} H\left(\boldsymbol{x}_t^*, \boldsymbol{p}_{t+1}^*, \boldsymbol{\theta}_t^*\right) \\ &= \boldsymbol{p}_{t+1}^* \cdot \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t^*, \boldsymbol{\theta}_t^*) + \nabla_{\boldsymbol{x}} L(\boldsymbol{x}_t^*, \boldsymbol{\theta}_t^*) \ , \end{aligned} \tag{22}$$

where $\boldsymbol{p}_t^*$ is the gradient of the total loss function w.r.t. the activation at layer t.

Finally, the maximization in (19) can be difficult to solve exactly since the dimension of the parameter is typically millions for DNNs. We can, however, apply approximated updates iteratively using first-order derivatives, i.e.

$$\boldsymbol{\theta}_t^{(i+1)} = \boldsymbol{\theta}_t^{(i)} - \eta \nabla_{\boldsymbol{\theta}_t} \mathbb{E}_{\mu_0} H(\boldsymbol{x}_t^*, \boldsymbol{p}_{t+1}^*, \boldsymbol{\theta}_t^{(i)}) \ . \tag{23}$$

The subscript t denotes the time step in the DNN dynamics, i.e. the index of the layer, whereas the superscript (i) represents the iterative update of the parameters in the outer loop. The update rule in (23) is equivalent to performing gradient descent on the original objective function J in (15),

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_t} H\left(\boldsymbol{x}_t^*, \boldsymbol{p}_{t+1}^*, \boldsymbol{\theta}_t\right) &= \boldsymbol{p}_{t+1}^* \cdot \nabla_{\boldsymbol{\theta}_t} f(\boldsymbol{x}_t^*, \boldsymbol{\theta}_t) + \nabla_{\boldsymbol{\theta}_t} L(\boldsymbol{x}_t^*, \boldsymbol{\theta}_t) \\ &= \nabla_{\boldsymbol{\theta}_t} J \ . \end{aligned} \tag{24}$$

When the expectation in (23) is replaced with a sample mean, E *et al.* [63] showed that if a solution of (16) is stable[5], we can find with high probability a random variable in its neighborhood that is a stationary solution of the sampled PMP.

3) Mean-Field Hamilton-Jacobi-Bellman Equation: The Hamilton-Jacobi-Bellman (HJB) equation [80] characterizes both necessary and sufficient conditions to the problem (15). The equation is derived from the principle of Dynamic Programming (DP), which reduces the problem of minimizing over a sequence of control to a sequence of minimization over a single control at each time. This is done by recursively solving the value function (define precisely below), and the obtained optimal policy is a function applied globally to the state space, i.e. a feedback controller with states as input.

E *et al.* [63] adapted the analysis to mean-field extension by considering probability measures as states. Following their derivations, we will consider the class of probability measures that is square integrable on Euclidean space with 2-Wasserstein metrics, denoted $\mathcal{P}_2(\mathbb{R})$, throughout our analysis. Importantly, this will lead to an infinite-dimensional HJB equation as we will show later. It is useful to begin with defining the cost-to-go and value function, denoted as J and v^*:

$$J(t,\mu,\boldsymbol{\theta}_t) := \mathbb{E}_{\substack{(\mathbf{x}_t,\mathbf{y})\sim\mu_t \\ \text{subject to (1)}}}\left[\Phi(\mathbf{x}_T,\mathbf{y}) + \int_t^T L(\mathbf{x}_t,\boldsymbol{\theta}_t)\,dt\right] \tag{25}$$

$$v^*(t,\mu) := \inf_{\boldsymbol{\theta}_t\in L^\infty} J(t,\mu,\boldsymbol{\theta}_t) \tag{26}$$

Note that the expectation is taken over the distribution evolution, starting from μ and propagating through the DNN architecture. The cost-to-go function is simply a generalization of the objective (16) to varying start time, and its infimum over the control space is achieved by the value function, i.e. the objective in (16) can be regarded as $J(0,\mu_0,\boldsymbol{\theta}_0)$, with $v^*(0,\mu_0)$ as its infimum. Now, we are ready to introduce the mean-field DP and HJB equation.

Theorem 2 (Mean-Field DP & HJB [63])**.** *Let the following statements be true:*

(A1') *f, L, Φ are bounded; f, L, Φ are Lipschitz continuous with respect to $\mathbf{x}_t$, and the Lipschitz constants of f and L are independent of $\boldsymbol{\theta}_t$.*

(A2') $\mu_0 \in \mathcal{P}_2\left(\mathbb{R}^{n+d}\right)$.

Then, both (25) $J(t,\mu,\boldsymbol{\theta}_t)$ and (26) $v^*(t,\mu)$ are Lipschitz continuous on $[0,T]\times\mathcal{P}_2\left(\mathbb{R}^{n+d}\right)$. For all $0\le t\le\hat{t}\le T$, the principle of dynamic programming suggests

$$v^*(t,\mu) = \inf_{\boldsymbol{\theta}_t\in L^\infty} \mathbb{E}_{\substack{(\mathbf{x}_t,\mathbf{y})\sim\mu_t \\ \textit{subject to (1)}}}\left[\int_t^{\hat{t}} L(\mathbf{x}_t,\boldsymbol{\theta}_t)\,dt + v^*\left(\hat{t},\hat{\mu}\right)\right], \tag{27}$$

where $\hat{\mu}$ denotes the terminal distribution at $\hat{t}$. Furthermore, Taylor expansion of the (27) gives the (28) equation,

$$\begin{cases} \partial_t v + \inf_{\boldsymbol{\theta}_t\in L^\infty}\langle \partial_\mu v(\mu)(\cdot), f(\cdot,\boldsymbol{\theta}_t)\rangle_\mu + \langle L(\cdot,\boldsymbol{\theta}_t)\rangle_\mu = 0, \\ v(T,\mu) = \langle\Phi(\cdot)\rangle_\mu, \end{cases} \tag{28}$$

[5] The mapping $F: U \mapsto V$ is said to be stable on $S_\rho(x) := \{y\in U : \|x-y\|_U \le \rho\}$ if $\|y-z\|_U \le K_\rho\|F(y)-F(z)\|_V$ for some $K_\rho > 0$.

where we recall $\langle f(\cdot),g(\cdot)\rangle_\mu := \int f(w)^\mathsf{T} g(w)\mathrm{d}\mu(w)$ and accordingly denote $\langle f(\cdot)\rangle_\mu := \int f(w)\mathrm{d}\mu(w)$. Finally, if $\boldsymbol{\theta}^*: (t,\mu)\mapsto\mathbb{R}^m$ is a feedback policy that achieves the infimum in (28) equation, then $\boldsymbol{\theta}^*$ is an optimal solution of the problem (16).

Notice that *(A1')* and *(A2')* are much stronger assumptions as opposed to those from Theorem 1. This is reasonable since the analysis is now adapted to take probability distributions as inputs. Theorem 2 differs from the classical HJB in that the equations become infinite-dimensional. The computation requires the derivative of the value function w.r.t. a probability measure, which can be done by recalling the definition of the *first-order variation* [81] of a function F at the probability measure μ, i.e. $\frac{\partial F(\mu)}{\partial\mu} \equiv \partial_\mu F$, satisfying the following relation:

$$F(\mu+\epsilon f) = F(\mu) + \epsilon\left\langle \partial_\mu F(\mu)(\cdot), f(\cdot)\right\rangle_\mu + \mathcal{O}(\epsilon^2)\ , \tag{29}$$

where ϵ is taken to be infinitesimally small. Note that $F(\cdot)$ and $\partial_\mu F(\mu)(\cdot)$ are functions respectively defined on the probability measure and its associated sample space. In other words, the derivative w.r.t. μ is achieved by interchanging probability measures with laws of random variables, to which we can apply a suitable definition of the derivative.

Due to the curse of dimensionality, classical HJB equations can be computationally intractable to solve for high-dimensional problems, let alone its mean-field extension. However, we argue that in the literature of DNN optimization, algorithms with a DP flavor, or at least an approximation of it, have not been well-explored. Research in this direction may provide a principled way to design feedback policies rather than the current open-loop solutions in which weights are fixed once the training ends. This can be particularly beneficial for problems related to e.g. adversarial attack and generalization analyses.

IV. Stochastic Optimization as a Dynamical System

We now turn attention to stochastic optimization. Recall that in Sec. III-C, we bridge optimization algorithms with controllers. In classical control theory, controllers alone can be characterized as separated dynamical systems. Stability analysis is conducted on the compositional system along with the plant dynamics [82]. Similarly, the dynamical perspective can be useful to study the training evolution and convergence property of DNN optimization. Unlike the deterministic propagation in Sec. III-A, stochasticity plays a central role in the resulting system due to the mini-batch sampling at each iteration. Preceding of time corresponds to the propagation of training cycles instead of forwarding through DNN layers. The stochastic dynamical viewpoint forms most of the recent study on SGD [14], [45], [70], [83].

This section is organized as follows. In Sec. IV-A, we will review the statistical property of the mini-batch gradient, which is the foundation for deriving the SGD dynamics. Recast of SGD as a continuous-time stochastic differential equation (SDE), or more generally a discrete-time master equation, will be demonstrated in Sec. IV-B, and IV-C, respectively.

The theoretical analysis from the dynamical framework consolidates several empirical observations, including implicit regularization on the loss landscape [11] and phase transition from a fast memorization period to slow generalization [18]. It can also be leveraged to design optimal adaptive strategies, as we will show in Sec. IV-D.

A. Preliminaries on Stochastic Mini-Batch Gradient

Slightly abuse the notation and denote the averaging training loss on the data set $\mathcal{D}$ as a function of parameter :

$$\Phi(\boldsymbol{\theta};\mathcal{D}) \equiv \Phi(\boldsymbol{\theta}) := \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} J(f(\mathbf{x}^{(i)},\boldsymbol{\theta}),\mathbf{y}^{(i)}) \ , \tag{30}$$

where J is the training objective for each sample (c.f. (15)) and $f \equiv f_0 \circ f_1 \circ f_2 \cdots$ includes all compositional functions of a DNN. We can write the full gradient of the training loss as $\nabla\Phi(\boldsymbol{\theta}) \equiv g(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} g^i(\boldsymbol{\theta})$, where $g^i(\boldsymbol{\theta})$ is the gradient on each data point $(\mathbf{x}^{(i)},\mathbf{y}^{(i)})$. The covariance matrix of $g^i(\boldsymbol{\theta})$, denoted $\boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta})$, is a positive-definite (P.D.) matrix which can be computed deterministically, given the DNN architecture, dataset, and current parameter, as

$$\begin{aligned}\text{Var}\left[g^i(\boldsymbol{\theta})\right] &:= \tfrac{1}{|\mathcal{D}|}\sum\nolimits_{i\in\mathcal{D}}\left(g^i(\boldsymbol{\theta})-g(\boldsymbol{\theta})\right)\left(g^i(\boldsymbol{\theta})-g(\boldsymbol{\theta})\right)^{\mathsf{T}} \\ &\equiv \boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta}) \ . \end{aligned} \tag{31}$$

Note that in practice, the eigen-spectrum of $\boldsymbol{\Sigma}_{\mathcal{D}}$ often features an extremely low rank ($< 0.5\%$ for both CIFAR-10 and CIFAR-100 as reported in [14]).

Access of $g(\boldsymbol{\theta})$ at each iteration is computationally prohibit for large-scale problems. Instead, gradients can only be estimated through a mini batch of i.i.d. samples $\mathcal{B} \subset \mathcal{D}$. When the batch size is large enough, $|\mathcal{B}| \gg 1$, CLT implies the mini-batch gradient, denoted $g^{mb}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} g^i(\boldsymbol{\theta})$, has the sample mean and covariance

$$\text{mean}\left[g^{mb}(\boldsymbol{\theta})\right] := \mathbb{E}_{\mathcal{B}}[g^{mb}(\boldsymbol{\theta})] \approx g(\boldsymbol{\theta}) \tag{32}$$

$$\begin{aligned}\text{Var}\left[g^{mb}(\boldsymbol{\theta})\right] :=& \mathbb{E}_{\mathcal{B}}[\left(g^{mb}(\boldsymbol{\theta})-g(\boldsymbol{\theta})\right)\left(g^{mb}(\boldsymbol{\theta})-g(\boldsymbol{\theta})\right)^{\mathsf{T}}] \\ \approx& \tfrac{1}{|\mathcal{B}|}\boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta}) \ . \end{aligned} \tag{33}$$

The last equality in (33) holds when $\mathcal{B}$ is sampled i.i.d. with replacement from $\mathcal{D}$.[6] For later purposes, let us also define the two-point noise matrix as $\tilde{\boldsymbol{\Sigma}}_{\mathcal{B}} := \mathbb{E}_{\mathcal{B}}[g^{mb}(\boldsymbol{\theta})g^{mb}(\boldsymbol{\theta})^{\mathsf{T}}]$. Its entry (i,j), or more generally the entry $(i_1,i_2,\cdots i_k)$ of a higher-order noise tensor, can be written as

$$\tilde{\Sigma}_{\mathcal{B},(i,j)} := \mathbb{E}_{\mathcal{B}}[g^{mb}(\theta_i)g^{mb}(\theta_j)] \text{ and} \tag{34}$$

$$\tilde{\Sigma}_{\mathcal{B},(i_1,i_2,\cdots i_k)} := \mathbb{E}_{\mathcal{B}}[g^{mb}(\theta_{i_1})g^{mb}(\theta_{i_2})\cdots g^{mb}(\theta_{i_k})] \ , \tag{35}$$

where $g^{mb}(\theta_i)$ is the partial derivative w.r.t. θ_i on the mini batch. Consequently, we can rewrite (33) as $\tilde{\boldsymbol{\Sigma}}_{\mathcal{B}} - g(\boldsymbol{\theta})g(\boldsymbol{\theta})^{\mathsf{T}}$.

Finally, we will denote the distribution of the parameter at training cycle t as $\rho_t(\boldsymbol{z}) := \rho(\boldsymbol{z},t) \propto \mathbb{P}(\boldsymbol{\theta}_t = \boldsymbol{z})$ and the steady-state distribution as $\rho^{\text{ss}} := \lim_{t\to\infty}\rho_t$.

[6] $\text{Var}\left[g^{mb}(\boldsymbol{\theta})\right] = \text{Var}\left[\frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}} g^i(\boldsymbol{\theta})\right] = \frac{1}{|\mathcal{B}|^2}\sum_{\mathcal{B}}\text{Var}\left[g^i(\boldsymbol{\theta})\right] = \frac{1}{|\mathcal{B}|}\boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta})$. The second equality holds since $\text{Cov}(g^i,g^j) = 0$ for $i \neq j$.

B. Continuous-time Dynamics of SGD

1) Derivation: The approximations from (32)-(33) allow us to replace $g^{mb}(\boldsymbol{\theta})$ with a Gaussian $\mathcal{N}\left(g(\boldsymbol{\theta}), \frac{1}{|\mathcal{B}|}\boldsymbol{\Sigma}_{\mathcal{D}}\right)$. The updated rule of SGD at each iteration can hence be recast as

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \eta g^{mb}(\boldsymbol{\theta}_t) \\ &\approx \boldsymbol{\theta}_t - \eta g(\boldsymbol{\theta}_t) + \tfrac{\eta}{\sqrt{|\mathcal{B}|}}\boldsymbol{\Sigma}_{\mathcal{D}}^{\frac{1}{2}}(\boldsymbol{\theta}_t)\mathbf{Z}_t \ , \end{aligned} \tag{36}$$

where η is the learning rate and $\mathbf{Z}_t \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. Now, consider the following continuous-time SDE and its Euler discretization,

$$\mathrm{d}\boldsymbol{\theta}_t = b(\boldsymbol{\theta}_t)\mathrm{d}t + \sigma(\boldsymbol{\theta}_t)\mathrm{d}\mathbf{W}_t \tag{37}$$

$$\Rightarrow \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + b(\boldsymbol{\theta}_t)\Delta t + \sqrt{\Delta t}\cdot\sigma(\boldsymbol{\theta}_t)\mathbf{Z}_t \ , \tag{38}$$

In the standard SDE analysis [84], $b(\boldsymbol{\theta}_t)$ and $\sigma(\boldsymbol{\theta}_t)$ refer to the drift and diffusion function. $\mathrm{d}\mathbf{W}_t$ is the Wiener process, or Brownian motion, in the same dimension of $\boldsymbol{\theta} \in \mathbb{R}^m$. It is easy to verify that (38) resembles (36) if we set $\Delta t \sim \eta$, $b(\boldsymbol{\theta}_t) \sim -g(\boldsymbol{\theta}_t)$, and $\sigma(\boldsymbol{\theta}_t) \sim \sqrt{\eta/|\mathcal{B}|}\boldsymbol{\Sigma}_{\mathcal{D}}^{\frac{1}{2}}$. We have therefore derived the continuous-time limit of SGD as the following SDE,

$$\mathrm{d}\boldsymbol{\theta}_t = -g(\boldsymbol{\theta}_t)\mathrm{d}t + \sqrt{2\beta^{-1}\boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta}_t)}\mathrm{d}\mathbf{W}_t \ , \tag{39}$$

where $\beta = \frac{2|\mathcal{B}|}{\eta}$ is proportional to the inverse temperature in thermodynamics.

Simply viewing (39) already gives us several insights. First, β controls the magnitude of the diffusion process. A similar relationship, named *noise scale*, between the batch size and learning rate has been proposed in [85], [86]. These two hyper-parameters, however, are not completely interchangeable since they contribute to different properties of the loss landscape [40]. Secondly, the stochastic dynamics is characterized by the drift term from the gradient descent flow $-g(\boldsymbol{\theta}_t)$ and the diffusion process from $\boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta}_t)$. When the parameter is still far from equilibrium, we can expect the drifting to dominate the propagation. As we approach flat local minima, fluctuations from the diffusion become significant. This drift-to-fluctuation transition has been observed on the Information Plane [18] and can be derived exactly for convex cases [87].

Since the two Wiener processes in (39) and (36) are independent, the approximation is valid only up to the distribution level. While a more accurate approximation is possible by introducing stochastic modified equations [88], we will limit the analysis to (39) and study the resulting dynamics using stochastic calculus and statistical physics.

2) Dynamics of Training Loss: To describe the propagation of the training loss, $\Phi(\boldsymbol{\theta}_t)$, as a function of the stochastic process in (39), we need to utilize Itô lemma, an extension of the chain rule in the ordinary calculus to the stochastic setting:

Lemma 1 (Itô lemma [89])**.** *Consider the stochastic process* $\mathrm{d}X_t = b(X_t,t)\mathrm{d}t + \sigma(X_t,t)\mathrm{d}W_t$. *Suppose* $b(\cdot,\cdot)$ *and* $\sigma(\cdot,\cdot)$ *follow appropriate smooth and growth conditions, then for a*

given function $V(\cdot,\cdot) \in C^{2,1}(\mathbb{R}^d \times [0,T])$, $V(X_t,t)$ is also a stochastic process:

$$\begin{aligned} \mathrm{d}V(X_t,t) = &\left[\partial_t V(X_t,t) + \nabla V(X_t,t)^\mathsf{T} b(X_t,t)\right]\mathrm{d}t \\ &+ \left[\frac{1}{2}\,\mathrm{Tr}\left[\sigma(X_t,t)^\mathsf{T} H_V(X_t,t)\sigma(X_t,t)\right]\right]\mathrm{d}t \\ &+ \left[\nabla V(X_t,t)^\mathsf{T}\sigma(X_t,t)\right]\mathrm{d}W_t\ , \end{aligned} \tag{40}$$

where $H_V(X_t,t)$ denotes the Hessian. i.e. $H_{V,(i,j)} = \partial^2 V/\partial x_i \partial x_j$.

Applying (40) to $V = \Phi(\boldsymbol{\theta}_t)$ readily yields the following SDE:

$$\begin{aligned} \mathrm{d}\Phi\left(\boldsymbol{\theta}_t\right) = &\left[-\nabla\Phi(\boldsymbol{\theta}_t)^\mathsf{T} g(\boldsymbol{\theta}_t) + \frac{1}{2}\,\mathrm{Tr}\left[\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}^{\frac{1}{2}} \boldsymbol{H}_\Phi \tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}^{\frac{1}{2}}\right]\right]\mathrm{d}t \\ &+ \left[\nabla\Phi(\boldsymbol{\theta}_t)^\mathsf{T}\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}^{\frac{1}{2}}\right]\mathrm{d}\mathbf{W}_t\ , \end{aligned} \tag{41}$$

where we denote $\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}^{\frac{1}{2}} = \sqrt{2\beta^{-1}\boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta}_t)}$ to simplify the notation. Taking the expectation of (41) over the parameter distribution $\rho_t(\boldsymbol{\theta})$ and recalling $\nabla\Phi = g$, the dynamics of the expected training loss can be described as

$$\mathrm{d}\mathbb{E}_{\rho_t}\left[\Phi(\boldsymbol{\theta}_t)\right] = \mathbb{E}_{\rho_t}\left[-\nabla\Phi^\mathsf{T}\nabla\Phi + \frac{1}{2}\,\mathrm{Tr}\left[\boldsymbol{H}_\Phi\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}\right]\right]\mathrm{d}t\ , \tag{42}$$

which is also known as the backward Kolmogorov equation [90], a partial differential equation (PDE) that describes the dynamics of a conditional expectation $\mathbb{E}[f(X_t)|X_s = x]$. Notice that $\mathrm{d}\mathbf{W}_t$ does not appear in (42) since the expectation of Brownian motion is zero. The term $\mathrm{Tr}[\boldsymbol{H}_\Phi\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}]$ draws a concrete connection between the noise covariance and the loss landscape. In [42], this trace quantity was highlighted as a measurement of the escaping efficiency from poor local minima. We can also derive the dynamics of other observables following similar steps.

When training converges, the left-hand side of (42) is expected to be near zero, i.e.

$$\mathbb{E}_{\rho^{\mathrm{ss}}}[\nabla\Phi^\mathsf{T}\nabla\Phi] \approx \frac{1}{2}\mathbb{E}_{\rho^{\mathrm{ss}}}[\mathrm{Tr}[\boldsymbol{H}_\Phi\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}]]\ . \tag{43}$$

In other words, the expected magnitude of the gradient signal is balanced off by the expected Hessian-covariance product measured in trace norm. A similar relation can be founded in the discrete-time setting (c.f. (57)), as we will see later in Sec. IV-C.

3) Dynamics of Parameter Distribution: The dynamics in (39) is a variant of the *damped Langevin diffusion*, which is widely used in statistical physics to model systems interacting with drag and random forces, e.g. the dynamics of pollen grains suspended in liquid, subjected to the friction from Navier-Stokes equations and random collisions from molecules. We synthesize the classical results for Langevin systems in the theorem below and discuss its implication in our settings.

Theorem 3 (Fokker-Plank equation and variational principle [91])**.** *Consider a damped Langevin dynamics with isotropic diffusion:*

$$\mathrm{d}X_t = -\nabla\Psi(X_t)\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}W_t\ , \tag{44}$$

where β is the damping coefficient. The temporal evolution of the probability density, $\rho_t \in C^{2,1}(\mathbb{R}^d \times \mathbb{R}^+)$, is the solution of the Fokker-Plank equation ((45)):

$$\partial_t\rho_t = \nabla\cdot(\rho_t\nabla\Psi) + \beta^{-1}\Delta\rho_t\ , \tag{45}$$

where $\nabla\cdot$, ∇ and Δ respectively denote the divergence, gradient, and Laplacian operators. Suppose Ψ is a potential function satisfying appropriate growth conditions, (45) has an unique stationary solution given by the Gibbs distribution $\rho^{\mathrm{ss}}(x;\beta) \propto \exp(-\beta\Psi(x))$. Furthermore, the stationary Gibbs distribution satisfies the variational principle — it minimizes the following functional

$$\rho^{\mathrm{ss}} = \arg\min_{\rho} \mathcal{F}_\Psi(\rho;\beta) := \mathcal{E}_\Psi(\rho) - \beta^{-1}\mathcal{S}(\rho)\ , \tag{46}$$

where $\mathcal{E}_\Psi(\rho) := \int_{\mathcal{X}}\Psi(x)\rho(x)\mathrm{d}x$ and $\mathcal{S}(\rho) := -\int_{\mathcal{X}}\rho(x)\log\rho(x)\mathrm{d}x$. In fact, $\mathcal{F}_\Psi(\rho_t;\beta)$ serves as a Lyapunov function for the (45), as it decreases monotonically along the dynamics of FPE and converges to its minimum, which is zero, at ρ^{ss}. In other words, we can rewrite (45) as a form of Wasserstein gradient flow (WGF):

$$\partial_t\rho_t = \nabla\cdot(\rho_t\nabla(\partial_\rho\mathcal{F}_\Psi))\ , \tag{47}$$

where $\partial_\rho\mathcal{F}_\Psi$ follows the same definition in (29), and is equal to $\log\frac{\rho}{\rho^{\mathrm{ss}}}+1$. We provide the derivation between (47) and (45) in Appendix B.

Equation (45) characterizes the deterministic transition of the density of an infinite ensemble of particles and is also known as the forward Kolmogorov equation [84]. The form of the Gibbs distribution at equilibrium reaffirms the importance of the temperature β^{-1}, as it determines the sharpness of ρ^{ss}. While high temperature can cause under-fitting, in the asymptotic limit as $\beta^{-1}\to 0$, the steady-state distribution will degenerate to point masses located at $\arg\max\Psi(x)$. From an information-theoretic viewpoint, $\mathcal{F}_\Psi(\rho;\beta)$ can be interpreted as the free energy, where $\mathcal{E}_\Psi(\rho)$ and $\mathcal{S}(\rho)$ are respectively known as the energy (or evidence) and entropy functionals. Therefore, minimizing $\mathcal{F}_\Psi$ balances between the likelihood of the observation and the diversity of the distribution.

Theorem 3 focuses on the isotropic diffusion process. Generalization to general diffusion is straightforward, and adapting the notations from our continuous limit of SGD in (39) yields

$$\partial_t\rho_t = \nabla\cdot\left(\nabla\Phi(\boldsymbol{\theta}_t)\rho_t + \beta^{-1}\nabla\cdot(\boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta}_t)\rho_t)\right)\ , \tag{48}$$

which is the (45) of the dynamics of the parameter distribution $\rho_t(\boldsymbol{\theta})$. Notice that when the analysis is lifted to the distribution space, the drift and diffusion are no longer separable as in (39).

Now, in order to apply the (46), $\Phi(\boldsymbol{\theta})$ needs to be treated as a potential function under the appropriate growth conditions, which is rarely the case under the setting of deep learning. Nevertheless, assuming these assumptions hold will lead to an important implication, suggesting that the implicit regularization stemmed from SGD can be mathematically quantified as entropy maximization. Another implication is that in the presence of non-isotropic diffusion during training[7],

[7] The non-isotropy of $\boldsymbol{\Sigma}_{\mathcal{D}}$ is expected since the dimension of the parameter is much larger than the number of data points used for training. Empirical supports can be founded in [14].

the trajectory governed by (48) has been shown to converge to a different location from the minima of the training loss [14]. In short, the variational inference implied from (39) takes the form

$$\arg\min_{\rho} \mathbb{E}_{\boldsymbol{\theta}\sim\rho_t}[\tilde{\Phi}(\boldsymbol{\theta})] - \beta^{-1}\mathcal{S}(\rho_t) \ , \tag{49}$$

which is minimized at $\rho^{\text{ss}}(\boldsymbol{\theta}) \propto \exp(-\beta\tilde{\Phi}(\boldsymbol{\theta}))$. The relationship between Φ and $\tilde{\Phi}$ at equilibrium is given by[8]

$$\nabla\Phi = \boldsymbol{\Sigma}_{\mathcal{D}}\nabla\tilde{\Phi} - \beta^{-1}\nabla\cdot\boldsymbol{\Sigma}_{\mathcal{D}} \ , \tag{50}$$

where the divergence $\nabla\cdot\boldsymbol{\Sigma}_{\mathcal{D}}$ is applied to the column space of the diffusion matrix. In other words, the critical points of $\tilde{\Phi}$ differ from those of the original training loss by the quantity $\beta^{-1}\nabla\cdot\boldsymbol{\Sigma}_{\mathcal{D}}$. It can be readily verified that $\tilde{\Phi} = \Phi$ if and only if $\boldsymbol{\Sigma}_{\mathcal{D}}$ is isotropic, i.e. $\boldsymbol{\Sigma}_{\mathcal{D}} = c\boldsymbol{I}_{\mathbb{R}^{\bar{m}}\times\mathbb{R}^{\bar{m}}}$ for some constant c. In fact, we can construct cases in which the most-likely trajectories traverse along closed loops, i.e. limit cycles, in the parameter space [14].

4) Remarks on Other SDE Modeling: We should be aware that (39), as a variant of the well-known Langevin diffusion, is only one of the possible realization of modeling stochastic mini-batch gradient. In fact, the metastability analysis of Langevin diffusion [92] conflicts with empirical observations in deep learning, as the analysis suggests an escape time depending exponentially on the depth of the loss landscape but only polynomial with its width. In other words, theoretical study implies Brownian-like processes should stay much longer in sharp local minima. To build some intuition on why this is true, recall that an implicit assumption we made when deriving (39) is the finite variance induced by $g^{mb}(\boldsymbol{\theta})$. Upper-bounding the second moment eventually prevents the presence of long-tail distributions, which plays a pivotal role in speeding up the exponential exit time of an SDE from narrow basins.

This issue has been mitigated in [68] by instead considering a general Lêvy process:

$$\mathrm{d}\boldsymbol{\theta}_t = -g(\boldsymbol{\theta}_t)\mathrm{d}t + \eta^{\frac{\alpha-1}{\alpha}}\sigma_\alpha(\boldsymbol{\theta}_t)\mathrm{d}L_t^\alpha \ , \tag{51}$$

where $\alpha\in(0,2]$ is the tail index and $\mathrm{d}L_t^\alpha$ denotes the α-stable Lêvy motion. The mini-batch gradients are now drawn from a zero-mean symmetric α-stable Lêvy distribution, $\mathcal{S}\alpha\mathcal{S}$-Levy$(0,\sigma_\alpha)$. Note that the moment of the distribution $\mathcal{S}\alpha\mathcal{S}$-Levy is finite up to only α order. When $\alpha = 2$, $\mathcal{S}\alpha\mathcal{S}$-Levy degenerates to a Gaussian and $\mathrm{d}L_t^\alpha$ is equivalent to a scaled Brownian motion. On the other hand, for $\alpha < 2$, the stochastic process in (51) features a Markov "jump" behavior, and theoretical study indicates a longer stay in, i.e. the process prefers, wider minima valleys [92]. The resulting heavy-tailed density aligns better with the empirical observation [14].

C. Discrete-time Dynamics of SGD

Despite the rich analysis by formulating SGD as a continuous-time SDE, we should remind us of the implicit assumptions for Itô-Stratonovich calculus to apply. Beside the smoothness conditions on the stochastic process, mini-batch gradients are replaced with Gaussian to bring Brownian motion into the picture. The fact that the mean square displacement of Brownian motion scales linearly in time, i.e. $\mathbb{E}[\mathrm{d}\mathbf{W}_t{}^2] = \mathrm{d}t$, leads to a quadratic expansion on the loss function, as shown in (41). In addition, the recast between (36) and (39) requires splitting η to $\sqrt{\eta}\sqrt{\mathrm{d}t}$. The continuous limit reached by sending $\mathrm{d}t\to 0^+$ while assuming finite $\sqrt{\eta}$ is arguably unjustified and pointed out in [55]. The authors instead proposed a discrete-time master equation that alleviates these drawbacks and is able to capture higher-order structures. We will restate the result and link it to the continuous-time SDE formulation as proceeding.

1) Derivation: Recall that $\rho_t(\boldsymbol{\theta})$ is the distribution of the parameter at time t. Its dynamics, when propagating to the next cycle $t+1$, can be written generally as

$$\rho_{t+1}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}'\sim\rho_t,\mathcal{B}}\left[\delta\left\{\boldsymbol{\theta} - \left[\boldsymbol{\theta}' - \eta g^{mb}(\boldsymbol{\theta}')\right]\right\}\right] \ , \tag{52}$$

where $\delta\{\cdot\}$ is the Kronecker delta and the expectation is taken over both the current distribution and the mini-batch sampling. Given an observable $\mathcal{O}(\boldsymbol{\theta}):\mathbb{R}^{\bar{m}}\mapsto\mathbb{R}$, its master equation at steady-state equilibrium ρ^{ss} can be written as

$$\mathbb{E}_{\rho^{\text{ss}}}[\mathcal{O}(\boldsymbol{\theta})] = \mathbb{E}_{\rho^{\text{ss}}}\left[\mathbb{E}_{\mathcal{B}}\left[\mathcal{O}\left(\boldsymbol{\theta} - \eta g^{mb}(\boldsymbol{\theta})\right)\right]\right] \ . \tag{53}$$

Full derivations are left in Appendix C. Note that the only assumption we make so far is the existence of ρ^{ss}. Equation (53) suggests that at equilibrium, the expectation of an observable remains unchanged when averaging over the stochasticity from mini-batch gradient updates.

2) Fluctuation Dissipation of Training Loss: Let we proceed by plugging the training loss $\Phi(\boldsymbol{\theta})$ to our observable of interest in (53). Taylor expanding it at $\eta = 0$ gives

$$\begin{aligned}
&\mathbb{E}_{\rho^{\text{ss}}}[\Phi(\boldsymbol{\theta})] \\
=&\mathbb{E}_{\rho^{\text{ss}}}\left[\mathbb{E}_{\mathcal{B}}\left[\Phi\left(\boldsymbol{\theta} - \eta g^{mb}(\boldsymbol{\theta})\right)\right]\right] \\
=&\mathbb{E}_{\rho^{\text{ss}}}\left[\Phi(\boldsymbol{\theta}) + \sum_{k=1}^{\infty}\frac{(-\eta)^k}{k!}\mathrm{D}^k\mathbb{E}_{\mathcal{B}}\left[\Phi\left(\boldsymbol{\theta} - \eta g^{mb}(\boldsymbol{\theta})\right)\right]\right] \ ,
\end{aligned} \tag{54}$$

where $\mathrm{D}^k F$ denotes the k-ordered expansion on a multivariate function F. Specifically, the first and second expansions can be written as[9]

$$\begin{aligned}
\mathrm{D}^1\mathbb{E}_{\mathcal{B}}\left[\Phi\left(\boldsymbol{\theta} - \eta g^{mb}(\boldsymbol{\theta})\right)\right] =& \textstyle\sum_i \partial_{\theta_i}\Phi\cdot\mathbb{E}_{\mathcal{B}}\left[g^{mb}(\theta_i)\right] \\
=&\nabla\Phi^{\mathsf{T}}\nabla\Phi
\end{aligned} \tag{55}$$

$$\begin{aligned}
\mathrm{D}^2\mathbb{E}_{\mathcal{B}}\left[\Phi\left(\boldsymbol{\theta} - \eta g^{mb}(\boldsymbol{\theta})\right)\right] =& \textstyle\sum_{(i,j)} H_{\Phi,(i,j)}\tilde{\boldsymbol{\Sigma}}_{\mathcal{B},(i,j)} \\
=&\operatorname{Tr}\left[\boldsymbol{H}_\Phi\tilde{\boldsymbol{\Sigma}}_{\mathcal{B}}\right] \ .
\end{aligned} \tag{56}$$

[8] The derivation of (50) is quite involved as it relies on the equivalence between Itô and A-type stochastic integration for the same FPE. We refer readers to [14] for a complete treatment.

[9] Applying the chain rule to $\mathrm{D}^k\mathbb{E}_{\mathcal{B}}[\cdots]$ in (54) yields a clean form as $\mathrm{D}^k\mathbb{E}_{\mathcal{B}}\left[\Phi\left(\boldsymbol{\theta} - \eta g^{mb}(\boldsymbol{\theta})\right)\right] = \sum\mathrm{D}^k_{(i_1,i_2,\cdots i_k)}\Phi(\boldsymbol{\theta})\cdot\tilde{\boldsymbol{\Sigma}}_{\mathcal{B},(i_1,i_2,\cdots i_k)}$, where we recall (35) and denote $\mathrm{D}^k_{(i_1,i_2,\cdots i_k)}$ as the k-ordered partial derivatives w.r.t. parameter indices $i_1,i_2,\cdots i_k$. For instance, $\mathrm{D}^1_{(i)}\Phi(\boldsymbol{\theta}) = \partial_{\theta_i}\Phi$ corresponds to the i-element of the full gradient. $\mathrm{D}^2_{(i,j)}\Phi(\boldsymbol{\theta}) = \partial^2\Phi/\partial\theta_i\partial\theta_j$ refers to the (i,j)-entry of the Hessian. The summation $\sum$ is taken over combinations of indices $i_1,i_2,\cdots i_k$.

For the last equality to hold in (54), the expectation of the infinite-series summation needs to vanish. Substituting (55-56) to (54), we will obtain the following relation

$$\mathbb{E}_{\rho^{\text{ss}}}\left[\nabla\Phi^{\mathsf{T}}\nabla\Phi\right] = \frac{\eta}{2}\mathbb{E}_{\rho^{\text{ss}}}\left[\text{Tr}\left[\boldsymbol{H}_{\Phi}\tilde{\boldsymbol{\Sigma}}_{\mathcal{B}}\right]\right] + \sum_{k=3}^{\infty}\frac{(-\eta)^k}{k!}\mathbb{E}_{\rho^{\text{ss}},\mathcal{B}}\left[\text{D}^k\Phi(\boldsymbol{\theta}-\eta g^{mb}(\boldsymbol{\theta}))\right] \ . \tag{57}$$

(57) can be viewed as the fluctuation-dissipation equation, a key concept rooted in statistical mechanics for bridging microscopic fluctuations to macroscopic dissipative phenomena [55]. It should be noted, however, that (57) is the necessary but not sufficient condition to ensure stationary.

Let us compare (57) with its continuous-time counterpart in (43). First, notice the difference between the two-point noise matrix, $\tilde{\boldsymbol{\Sigma}}_{\mathcal{B}}$, and the covariance matrix of the sample gradient, $\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}$. In fact, these two matrices can be related by

$$\tilde{\boldsymbol{\Sigma}}_{\mathcal{B}} \approx \left(\tfrac{1}{|\mathcal{B}|}-\tfrac{1}{|\mathcal{D}|}\right)\boldsymbol{\Sigma}_{\mathcal{D}} = \tfrac{1}{\eta}\left(1-\tfrac{|\mathcal{B}|}{|\mathcal{D}|}\right)\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}} \approx \tfrac{1}{\eta}\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}} \ , \tag{58}$$

where the first approximation is followed by Proposition 1 in [93], and the second one by assuming $|\mathcal{B}| \ll |\mathcal{D}|$. In the small learning rate regime, (43) and (57) are essentially equivalent and we consolidate the analysis from the continuous-time framework in Sec. IV-B. The higher-order terms in (57) measure the anharmonicity of the loss landscape, which becomes nontrivial when the learning rate is large.

D. Improving SGD with Optimal Control

Interpreting SGD as a stochastic dynamical system makes control theory applicable. Note that this is different from what we have derived in Sec. III. The state space on which we wish to impose control is the parameters space $\mathbb{R}^{\bar{m}}$, instead of the activation space $\mathbb{R}^n$. The fact that in Sec. III-C we are managing to apply control in $\mathbb{R}^{\bar{m}}$ limits out capability to go beyond theoretical characterization to practical algorithmic design due to high dimensionality. In contrast, here we do not specify where the control should take place, depending on how we introduce it to the stochastic dynamical system. Such a flexibility has led to algorithmic improvement of SGD dynamics. For instance, using optimal control to derive optimal adaptive strategies for hyper-parameters [69].

The literature on adaptive (e.g. annealed) learning rate scheduling has been well-studied for convex problems [87], [94]. The heuristic of decaying the learning rate with training cycle, i.e. $\sim 1/t$, typically works well for DNN training, despite its non-convexity. From the optimal control perspective, we can formulate the scheduling process mathematically by introducing to (39) a rescaling factor $u_t \in (0,1]$ and its continuous-time function $u_{t\to T} : [t,T] \mapsto (0,1]$. Applying similar derivations using Itô Lemma, we obtain

$$\text{d}\boldsymbol{\theta}_t = -u_t g(\boldsymbol{\theta}_t)\text{d}t + u_t\sqrt{2\beta^{-1}\boldsymbol{\Sigma}_{\mathcal{D}}(\boldsymbol{\theta}_t)}\text{d}\mathbf{W}_t \ , \tag{59}$$

$$\text{d}\mathbb{E}\left[\Phi(\boldsymbol{\theta}_t)\right] = \mathbb{E}\left[-u_t\nabla\Phi^{\mathsf{T}}\nabla\Phi + \frac{u_t^2}{2}\text{Tr}\left[\boldsymbol{H}_{\Phi}\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}\right]\right]\text{d}t. \tag{60}$$

The *stochastic optimal control problem* from this new dynamics can be written as

$$\min_{u_{t\to T}} \mathbb{E}\left[\Phi(\boldsymbol{\theta}_T)\right] \quad \text{s.t. (60)} \ , \tag{61}$$

where T is the maximum training cycle. Li *et al.* [69] showed that when $\Phi(\boldsymbol{\theta})$ is *quadratic*, solving (61) using the HJB equation (recall Theorem 2) yields a closed-form policy

$$u_t^* = \min(1, \frac{\mathbb{E}[\Phi(\boldsymbol{\theta}_t)]}{\eta\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}}) \ . \tag{62}$$

Intuitively, this optimal strategy suggests using the maximum learning rate when far from minima and decay it whenever fluctuations begin to dominate. Further expansion on the ratio $\mathbb{E}[\Phi(\boldsymbol{\theta}_t)]/\eta\tilde{\boldsymbol{\Sigma}}_{\mathcal{D}}$ will give us the annealing schedule of $\mathcal{O}(1/t)$. In other words, the strategy proposed in the previous study is indeed optimal from the optimal control viewpoint. Also, notice that (62) is a feedback policy since the optimal adaptation depends on the statistics of the current parameter.

For general loss functions, (62) can serve as a well-motivated heuristic. The resulting scheduling scheme has been shown to be more robust to initial conditions when compared with other SGD variants [69]. Similarly, we can derive optimal adaptation strategies for other hyper-parameters, such as the momentum and batch size [69], [70]. Lastly, we note that other learning rate adaptations, such as the constant-and-cut scheme, can also be included along this line by modifying (59) to accept general Markov jump processes.

V. Beyond Supervised Learning

The optimal control framework in Sec. III-C fits with supervised learning by absorbing labels into the terminal cost or augmented state space. In this section, we demonstrate how to extend the framework to other learning problems. Specifically, by allowing standard (i.e. risk-neutral) (15) and (16) objectives, which minimize the expected loss incurred from the stochasticity, to be risk-aware, we generalize the formulation to consider statistical behaviors from higher-order moments. Depending on the problem setting, the risk-aware optimal control problem can be recast to Bayesian learning and adversarial training, as we will show in Sec. V-B and V-C. While the former viewpoint has been leveraged to impose priors on the training dynamics [44], [45], the latter seeks to optimize worst-case perturbations from an adversarial attacker. Additionally, we will interpret meta-learning algorithms with a specific structure as feedback controllers in Sec. V-D.

A. Preliminaries on Risk-Aware Optimal Control

Risk sensitivity has been widely used in Markovian decision processes (MDPs) that require more sophisticated criteria to reflect the variability-risk features of the problems [95]. The resulting optimal control framework is particularly suitable for stochastic dynamical systems and closely related to robust and minimax control [96]. To bring risk awareness into the original training objective, i.e. the per-sample objective J in (15), we need to consider the following generalized exponential utility function:

$$\mathcal{J}_k(\boldsymbol{x},\xi) := \begin{cases} \frac{1}{k}\log\left\{\mathbb{E}_{\xi}\left[\exp\left(kJ(\boldsymbol{x},\xi)\right)\right]\right\} & , k\neq 0 \\ \mathbb{E}_{\xi}\left[J(\boldsymbol{x},\xi)\right] & , k=0 \end{cases} \ , \tag{63}$$

where ξ denotes any source of stochasticity that is being averaged over the expectation. When $k=0$, $\mathcal{J}_k$ reduces to

the risk-neutral empirical mean, i.e. $\Phi(\boldsymbol{\theta})$ in (30). In contrast, the log partition functional for $k \neq 0$ has a risk-aware interpretation, which can be mathematically described as

$$\mathcal{J}_{k\neq 0}(\boldsymbol{x},\xi) \approx \mathbb{E}_\xi J + \frac{k}{2}\mathrm{Var}_\xi\left[J\right] \ . \tag{64}$$

We left the full derivation in Appendix D. For positive k, the objective is *risk-averse* since in addition to the expectation, we also penalize the variation of the loss. In contrast, $k < 0$ results in a *risk-seeking* behavior as we now favor higher variability. From the optimization viewpoint, the log partition functional can be thought of as an approximation of a smooth $\max$ operator. The objective in (63) therefore inherits an inner-loop $\max$/$\min$ optimization, depending on the sign of k:

$$\min_{\boldsymbol{x}} \mathcal{J}_{k\neq 0}(\boldsymbol{x},\xi) \approx \begin{cases} \min\limits_{\boldsymbol{x}} \max\limits_{\xi} J(\boldsymbol{x},\xi) & \text{if} \quad k>0 \\ \min\limits_{\boldsymbol{x}} \min\limits_{\xi} J(\boldsymbol{x},\xi) & \text{if} \quad k<0 \end{cases} \tag{65}$$

From such, it is handy to characterize the optimal policy of a min-max objective as risk-averse, whereas the one from a min-min objective often reveals a risk-seeking tendency. This interpretation will become useful as we proceed to Sec. V-B and V-C.

B. Bayesian Learning & Risk-Seeking Control

Recall that flatter minima enjoy lower generalization gap since they are less sensitive to perturbations of the data distribution [11]. In this spirit, Chaudhari *et al.* [44] proposed the following *local entropy loss* in order to guide the SGD dynamics towards flat plateaus faster:

$$\Phi_{\text{ent}}(\boldsymbol{\theta};\gamma) := -\log \int_{\boldsymbol{\theta}'\in\mathbb{R}^m} \exp(-\Phi(\boldsymbol{\theta}') - \frac{\gamma}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_2^2)\mathrm{d}\boldsymbol{\theta}' \ , \tag{66}$$

where $\Phi(\cdot)$ is defined in (30) and the hyper-parameter γ controls the degree of trade-off between the depth and width of the loss landscape. This surrogate loss is well-motivated from the statistical physics viewpoint since the objective balances between an energetic term (i.e. training loss) and entropy term (i.e. flatness of local geometry). In addition, it can be connected to numerical analysis on nonlinear PDE [45]. Here, we provide an alternative perspective from risk-aware control and its connection to Bayesian inference.

We know from Sec. V-A that the log partition functional approximates the $\max$ operator. Minimizing the local entropy loss therefore becomes

$$\begin{aligned} \min_{\boldsymbol{\theta}} \Phi_{\text{ent}}(\boldsymbol{\theta};\gamma) &\approx \min_{\boldsymbol{\theta}} -\max_{\boldsymbol{\theta}'} \left\{-\Phi(\boldsymbol{\theta}') - \frac{\gamma}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_2^2\right\} \\ &= \min_{\boldsymbol{\theta}} \min_{\boldsymbol{\theta}'} \left\{\Phi(\boldsymbol{\theta}') + \frac{\gamma}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_2^2\right\} \ , \end{aligned} \tag{67}$$

which is a nested optimization with an inner loop minimizing the same loss with a regularization term centered at $\boldsymbol{\theta}$. For fixed $\boldsymbol{\theta}'$, the outer loop simply optimizes a locally-approximated quadratic $\frac{\gamma}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_2^2$. Casting this quadratic regularization as a distribution density and recalling the risk-seeking interpretation of the min-min objective, we have

$$\min_{\boldsymbol{\theta}} \Phi_{\text{ent}}(\boldsymbol{\theta};\gamma) \approx \min_{\boldsymbol{\theta}} \mathbb{E}_{\mathcal{P}_{\gamma,\boldsymbol{\theta}}}\left[\Phi(\boldsymbol{\theta}')\right] - \frac{1}{2}\mathrm{Var}_{\mathcal{P}_{\gamma,\boldsymbol{\theta}}}\left[\Phi(\boldsymbol{\theta}')\right] \ . \tag{68}$$

$\mathcal{P}_{\gamma,\boldsymbol{\theta}}$ denotes the Gibbs distribution, $\mathcal{P}(\boldsymbol{\theta}';\gamma,\boldsymbol{\theta}) \propto Z^{-1}\exp(-\frac{\gamma}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_2^2)$, with Z^{-1} as the normalization term. The risk-seeking objective in (68) encourages exploration on areas with higher variability. This implies a potential improvement on the convergence speed, despite the overhead incurred from additional minimization.

Solving (67) requires an expensive inner-loop minimization over the entire parameter space at each iteration. This can, however, be estimated with the stochastic gradient Langevin dynamic [97], an MCMC sampling technique for Bayesian inference. The resulting algorithm, named *Entropy-SGD* [44], obeys the following dynamics:

$$\mathrm{d}\boldsymbol{z}_s = -\left[g^{mb}(\boldsymbol{z}_s) + \gamma\left(\boldsymbol{z}_s - \boldsymbol{\theta}_t\right)\right]\mathrm{d}s + \sqrt{\epsilon}\mathrm{d}\mathbf{W}_s \ , \tag{69}$$

$$\mathrm{d}\boldsymbol{\theta}_t = \gamma\left(\boldsymbol{z}-\boldsymbol{\theta}_t\right)\mathrm{d}t \ , \tag{70}$$

where $\boldsymbol{z}$ takes the same space as $\boldsymbol{\theta} \in \mathbb{R}^{\bar{m}}$ with the initial condition $\boldsymbol{z}_0 = \boldsymbol{\theta}_t$. Notice that the two dynamical systems in (69) and (70) operate in different time scales, denoted $\mathrm{d}s$ and $\mathrm{d}t$ respectively, and they correspond to the first-order derivatives of the inner and outer minimization in (67). Chaudhari *et al.* [45] showed that in the asymptotic limit of the non-viscous approximation, i.e. $\epsilon \to 0$, gradient descent on the local entropy loss, $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta\nabla_{\boldsymbol{\theta}}\Phi_{\text{ent}}(\boldsymbol{\theta}_t;\gamma)$, is equivalent to a *forward* Euler step on the original loss function, $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta\nabla_{\boldsymbol{\theta}}\Phi(\boldsymbol{\theta}_{t+1})$. In other words, we may interpret the dynamics in (69) as a one-step prediction (in a Bayesian fashion with a quadratic prior) of the gradient at the next iteration.

C. Adversarial Training as Minimax Control

Study on adversarial properties of DNN has become increasingly popular since the landmark paper in [98] revealed its vulnerability to human-invisible perturbations. The consequence can be catastrophic from a security standpoint [99], when machine learning algorithms in real-world applications, e.g. perception systems on self-driving vehicles, are intentionally fooled (i.e. attacked) to make incorrect decisions at test time. Among the attempts to robustify deep models, adversarial training proposes to solve the following optimization problem:

$$\min_{\boldsymbol{\theta}} \max_{\|\delta\|_p\leq\Delta} \Phi_{\text{adv}}(\boldsymbol{\theta},\delta) := \frac{1}{\mid\mathcal{D}\mid}\sum_{i\in\mathcal{D}} J(f(\mathbf{x}^{(i)}+\delta^{(i)},\boldsymbol{\theta}),\mathbf{y}^{(i)}) \ . \tag{71}$$

$\Phi_{\text{adv}}(\boldsymbol{\theta},\delta)$ is equivalent to the original training loss (c.f. (30)) subjected to sample-wise perturbations $\delta^{(i)}$, which are of the same dimension as the input space and constrained within a p-norm ball with radius Δ. Essentially, adversarial training seeks to find a minimizer of the worse-case performance when data points are adversarially distorted.

The min-max objective in (71) implies a risk-averse behavior, in contrast to the risk-seeking in (68). Classical analyses from the minimax control theory suggest a slow convergence and a conservative optimal policy. These arguments agree with practical observations as adversarial learning usually takes much longer time to train and admits a trade-off between adversarial robustness (e.g. proportion of data points that are adversarial) and generalization performance [99].

Algorithmically, the inner maximization is often evaluated on a set of adversarial examples generated on the fly, depending on the current parameter, and the adversarial training objective is replaced with a mixture of the original loss function and this adversarial surrogate. The min-max problem in (71) is hence lower-bounded by

$$\min_{\boldsymbol{\theta}} \max_{\|\delta\|_p \leq \Delta} \Phi_{\text{adv}}(\boldsymbol{\theta}, \delta) \geq \min_{\boldsymbol{\theta}} \alpha \Phi(\boldsymbol{\theta}) + (1-\alpha)\Phi_{\text{adv}}(\boldsymbol{\theta}, \hat{\delta}) \ , \tag{72}$$

where $\alpha \in (0, 1]$ is the mixture ratio and $\hat{\delta} := \text{Proj}_{\|\cdot\|_p \leq \Delta}[\text{Alg}(\boldsymbol{\theta}, \Phi(\cdot); \mathcal{D})]$ denotes the p-norm projected perturbation generated from an algorithm, Alg. The approach can be viewed as an adaptive data augmentation technique. We should note, however, that the i.i.d. assumption on the training dataset no longer holds in this scenario and we may require exponentially more data to prevent over-fitting [100]. Lastly, it is possible to instead upper-bound the objective with a convex relaxation, which will lead to a provably robust model to any norm-bounded adversarial attack [101].

D. Meta Learning as Feedback Controller

Meta-learning aims to discover prior knowledge from a set of learnable tasks such that the learned initial parameter provides a favorable inductive bias for fast adaptation to unseen tasks at test time. The learning problems, often called *learning to learn*, is naturally applicable to those involving prior distillation from limited data, such as few-shot classification [102] and reinforcement learning in fast-changing environments [103]. It can also be cast to probabilistic inference in a hierarchical Bayesian model [104]. Here, we bridge a popular branch of algorithms, namely *model-agnostic meta-learning* (MAML) [102], to the feedback control viewpoint.

In the problem formulation of MAML, an agent is given a distribution of tasks, $\mathcal{T}_i \sim \mathcal{P}_{\mathcal{T}}$, with the task-dependent cost function, $\Phi_{\mathcal{T}_i}(\cdot)$, and asked to find an initialization that can continuously adapt to other unseen tasks drawn from $\mathcal{P}_{\mathcal{T}}$. The meta-training objective and adaptation rule can be written as

$$\Phi_{\text{meta}}(\boldsymbol{\theta}; \mathcal{P}_{\mathcal{T}}) := \mathbb{E}_{\mathcal{T}_i \sim \mathcal{P}_{\mathcal{T}}} \left[\Phi_{\mathcal{T}_i}(\boldsymbol{\theta}^N_{\text{adapt}}) \right], \text{ where} \tag{73}$$

$$\boldsymbol{\theta}^{n+1}_{\text{adapt}} = \boldsymbol{\theta}^n_{\text{adapt}} - \bar{\eta} \nabla_{\boldsymbol{\theta}} \Phi_{\mathcal{T}_i}(\boldsymbol{\theta}^n_{\text{adapt}}) \quad \text{and} \quad \boldsymbol{\theta}^0_{\text{adapt}} = \boldsymbol{\theta} \ . \tag{74}$$

$\boldsymbol{\theta}^N_{\text{adapt}}$ denotes an N-step adaptation from the current parameter using gradient descent with the step size $\bar{\eta}$ at each update. N is a hyper-parameter that generalizes standard objectives to $\Phi_{\text{meta}}(\cdot)$ for positive N. As N increases, regularization will be imposed on the meta-training process in the sense that the agent is encouraged to find a minimizer no more than N steps away from the local minima of each task, instead of over-fitting to the one of any particular task.

Now, recall the interpretation of (13) as (14) in Sec. III-C. Through this lens, the adaptation rule in (74) can be thought of as an N-step integral controller, and minimizing (73) is equivalent to searching an optimal initial condition for the controller. Since feedback controllers are originally designed for problems requiring on-line adaptation and integral controllers feature zero steady-state errors, we consolidate the theoretical foundation of MAML-inspired algorithms. Implications from this viewpoint can leverage knowledge from control literature to design more sophisticated and/or principled adaptation rules. We may also derive optimal adaptation rules for other hyper-parameters, such as the step size $\bar{\eta}$ and adaptation number N, similar to what we have shown in Sec. IV-D.

VI. Conclusion

This review aims to align several seemly disconnected viewpoints of deep learning theory with the line of dynamical system and optimal control. We first observe that the compositionality of DNNs and the descending update in SGD suggest an interpretation of discrete-time (stochastic) dynamical systems. Rich mathematical analysis can be applied when certain assumptions are made to bring the realization to its continuous-time limit. The framework forms the basis of most recent understandings of deep learning, by recasting DNN as an ordinary differential equation and SGD as a stochastic differential equation. Among the available mathematical tools, we should highlight the significance of mean-field theory and stochastic calculus, which enable characterization of the dynamics of deep representation and stochastic functionals (e.g. the training loss or parameter distribution) at the ensemble level. The dynamical perspective alone has revealed valuable implications, as it successfully gives predictions to e.g. the trainability of random networks from critical initialization, the interaction between gradient drift and noise diffusion during training, the concrete form of implicit regularization from SGD, and even the global optimality of deep learning problems, to name a few.

Another appealing implication, despite receiving little attention, is to introduce the optimal control theory to the corresponding dynamics. To emphasize its importance, we note that the celebrated back-propagation algorithm is, in fact, an approximation of the Pontryagins Minimum Principle (PMP), a well-known theory dated back to the 1960s that describes the necessary conditions to the optimal control problems. Limited works inspired from this viewpoint include optimal adaptive strategies for hyper-parameters and minimum principle based optimization algorithms. When the standard optimal control objective is extended to accept higher-order statistical moments, the resulting "risk-aware" optimal control framework generalizes beyond supervised learning, to include problems such as Bayesian learning, adversarial training, and meta learning. We wish this article stands as a foundation to open up new avenues that may bridge and benefit communities from both deep learning and optimal control.

For future directions, we note that the optimal control theory for DNNs training is far from being completed, and relaxing the currently presented theorems to a more realistic setting will be beneficial. For instance, despite the thorough discussion in Sec. III-C, our derivation is mainly constructed upon the continuous-time framework to avoid the difficulties incurred from the discrete-time analysis. Additionally, while an initial attempt to bridge other learning problems to the proposed framework has been taken in Sec. V, more are left to be explored. Specifically, Generative Adversarial Networks are closely related to minimax control, and the dynamical analysis

from an SDE viewpoint has been recently discussed to reveal an intriguing variational interpretation [105].

APPENDIX A

Critical initialization and mean field approximation can be applied to convolution layers as the number of channels goes to limit [27]. Similar results can be derived except $\boldsymbol{\epsilon}_t$ now traverses with a much richer dynamics through convoluted operators. For recurrent architectures, e.g. RNN and LSTM, the theory suggests that gating mechanisms facilitate efficient signal propagation [25], [26]. However, it also casts doubt on several practically-used modules, as the analysis suggests batch normalization causes exploding gradient signal [24] and dropout destroys the order-to-chaos critical point [106]. Global optimality of GD for other architectures can be proved following similar derivations. Appendix E in [19] provides a general framework to include FC-DNN, ResNet, and convolution DNN.

APPENDIX B

First, we notice that the variational functional $\mathcal{F}_\Psi(\rho;\beta)$ in (47) can be written as an KullbackLeibler divergence:

$$\mathcal{F}_\Psi(\rho;\beta) := \mathcal{E}_\Psi(\rho) - \beta^{-1}\mathcal{S}(\rho) = \beta^{-1} D_{\mathrm{KL}}(\rho||\rho^{\mathrm{ss}}) \ ,$$

where $\rho^{\mathrm{ss}} \propto \exp(-\beta\Psi(x))$. Now, recall that for a functional $\mathcal{F}: \mathcal{P}_2 \mapsto \mathbb{R}$ of the following form: $\mathcal{F}(\rho) := \int_x f(\rho(x))\mathrm{d}x$, its first variation at ρ is given by $\partial_\rho \mathcal{F}(\rho)(\cdot) = f'(\rho(\cdot))$. Substituting it into (47) will lead to (45):

$$\begin{aligned}
\partial_t \rho_t =& \nabla\cdot(\rho_t \nabla(\partial_\rho \mathcal{F}_\Psi)) \\
=& \beta^{-1}\nabla\cdot(\rho_t\nabla(\partial_\rho D_{\mathrm{KL}}(\rho_t||\rho^{\mathrm{ss}}))) \\
=& \beta^{-1}\nabla\cdot\left(\rho_t\nabla\left(\frac{\mathrm{d}(\rho_t\log\frac{\rho_t}{\rho^{\mathrm{ss}}})}{\mathrm{d}\rho}\right)\right) \\
=& \beta^{-1}\nabla\cdot\left(\rho_t\nabla\left(\log\frac{\rho_t}{\rho^{\mathrm{ss}}} + \cancel{\rho_t}\frac{1}{\cancel{\rho_t}}\right)\right) \\
=& \beta^{-1}\nabla\cdot(\rho_t\nabla(\beta\Psi + \log\rho_t)) \\
=& \nabla\cdot(\rho_t\nabla\Psi) + \beta^{-1}\nabla\cdot\left(\rho_t\frac{\nabla\rho_t}{\rho_t}\right) \\
=& \nabla\cdot(\rho_t\nabla\Psi) + \beta^{-1}\Delta\rho_t
\end{aligned}$$

APPENDIX C

Here we recapitulate the derivation from [55]. First, recall (52):

$$\begin{aligned}
\rho_{t+1}(\boldsymbol{\theta}) =& \mathbb{E}_{\boldsymbol{\theta}'\sim\rho_t,\mathcal{B}}\left[\delta\left\{\boldsymbol{\theta} - \left[\boldsymbol{\theta}' - \eta g^{mb}(\boldsymbol{\theta}')\right]\right\}\right] \\
=& \mathbb{E}_{\mathcal{B}}\left[\int \mathrm{d}\boldsymbol{\theta}'\rho_t(\boldsymbol{\theta}')\delta\left\{\boldsymbol{\theta} - \left[\boldsymbol{\theta}' - \eta g^{mb}(\boldsymbol{\theta}')\right]\right\}\right] \ .
\end{aligned}$$

The steady-state distribution therefore obeys the relation

$$\rho^{\mathrm{ss}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathcal{B}}\left[\int \mathrm{d}\boldsymbol{\theta}'\rho^{\mathrm{ss}}(\boldsymbol{\theta}')\delta\left\{\boldsymbol{\theta} - \left[\boldsymbol{\theta}' - \eta g^{mb}(\boldsymbol{\theta}')\right]\right\}\right] . \tag{75}$$

Now, substitute (75) to the expectation of an observable $\mathcal{O}(\boldsymbol{\theta})$ at equilibrium

$$\begin{aligned}
&\mathbb{E}_{\rho^{\mathrm{ss}}}\left[\mathcal{O}(\boldsymbol{\theta})\right] \\
=& \int \mathrm{d}\boldsymbol{\theta}\rho^{\mathrm{ss}}(\boldsymbol{\theta})\mathcal{O}(\boldsymbol{\theta}) \\
=& \mathbb{E}_{\mathcal{B}}\left[\int \mathrm{d}\boldsymbol{\theta}\int \mathrm{d}\boldsymbol{\theta}'\rho^{\mathrm{ss}}(\boldsymbol{\theta}')\delta\left\{\boldsymbol{\theta} - \left[\boldsymbol{\theta}' - \eta g^{mb}(\boldsymbol{\theta}')\right]\right\}\mathcal{O}(\boldsymbol{\theta})\right] \\
=& \mathbb{E}_{\mathcal{B}}\left[\int \mathrm{d}\boldsymbol{\theta}\int \mathrm{d}\boldsymbol{\theta}'\rho^{\mathrm{ss}}(\boldsymbol{\theta}')\mathcal{O}(\boldsymbol{\theta}' - \eta g^{mb}(\boldsymbol{\theta}'))\right] \\
=& \mathbb{E}_{\rho^{\mathrm{ss}}}\left[\mathbb{E}_{\mathcal{B}}\left[\mathcal{O}\left(\boldsymbol{\theta} - \eta g^{mb}(\boldsymbol{\theta})\right)\right]\right] \ .
\end{aligned} \tag{76}$$

We obtain its master equation at equilibrium (53).

APPENDIX D

Recall the Taylor expansion of exp and log functions are $\exp(x) = 1 + \sum_{k=1}^{\infty}\frac{x^k}{k!}$, and $\log(1+x) = \sum_{k=1}^{\infty}(-1)^{k-1}\frac{x^k}{k}$. Expanding the objective $\mathcal{J}_{k\neq 0}$ up to second order leads to

$$\begin{aligned}
&\mathcal{J}_{k\neq 0}(\boldsymbol{x},\xi) \\
=& \frac{1}{k}\log\mathbb{E}_\xi\left[\exp\left(kJ\right)\right] \\
\approx& \frac{1}{k}\log\mathbb{E}_\xi\left[1 + kJ + \frac{k^2}{2}J^2\right] \\
\approx& \frac{1}{k}\left[\left(k\mathbb{E}_\xi J + \frac{k^2}{2}\mathbb{E}_\xi J^2\right) - \frac{1}{2}\left(k\mathbb{E}_\xi J + \frac{k^2}{2}\mathbb{E}_\xi J^2\right)^2\right] \\
=& \frac{1}{k}\left[k\mathbb{E}_\xi J + \frac{k^2}{2}\left[\mathbb{E}_\xi J^2 - (\mathbb{E}_\xi J)^2\right] + \mathcal{O}(k^3)\right] \\
=& \mathbb{E}_\xi J + \frac{k}{2}\mathrm{Var}_\xi\left[J\right] + \mathcal{O}(k^2) \ .
\end{aligned} \tag{77}$$

For small k, the higher-order term $\mathcal{O}(k^2)$ is negligible and we obtain the risk-aware interpretation of (64).

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[4] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231*, 2018.

[5] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *2016 international joint conference on neural networks (IJCNN)*. IEEE, 2016, pp. 4368–4374.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[7] A. Athalye and N. Carlini, "On the robustness of the cvpr 2018 white-box adversarial example defenses," *arXiv preprint arXiv:1804.03286*, 2018.

[8] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[9] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, "Exponential expressivity in deep neural networks through transient chaos," in *Advances in neural information processing systems*, 2016, pp. 3360–3368.

[10] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in neural information processing systems*, 2014, pp. 2933–2941.

[11] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[12] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[13] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," *arXiv preprint arXiv:1509.01240*, 2015.

[14] P. Chaudhari and S. Soatto, "Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks," in *2018 Information Theory and Applications Workshop (ITA)*. IEEE, 2018, pp. 1–10.

[15] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[16] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht, "First-order methods almost always avoid saddle points," *arXiv preprint arXiv:1710.07406*, 2017.

[17] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[18] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," *arXiv preprint arXiv:1703.00810*, 2017.

[19] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," *arXiv preprint arXiv:1811.03804*, 2018.

[20] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*, 2018.

[21] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, "On the information bottleneck theory of deep learning," in *International Conference on Learning Representations*, 2018.

[22] D. Zou, Y. Cao, D. Zhou, and Q. Gu, "Stochastic gradient descent optimizes over-parameterized deep relu networks," *arXiv preprint arXiv:1811.08888*, 2018.

[23] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," *arXiv preprint arXiv:1811.03962*, 2018.

[24] S. S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein, "Deep information propagation," *arXiv preprint arXiv:1611.01232*, 2016.

[25] M. Chen, J. Pennington, and S. S. Schoenholz, "Dynamical isometry and a mean field theory of rnns: Gating enables signal propagation in recurrent neural networks," *arXiv preprint arXiv:1806.05394*, 2018.

[26] D. Gilboa, B. Chang, M. Chen, G. Yang, S. S. Schoenholz, E. H. Chi, and J. Pennington, "Dynamical isometry and a mean field theory of lstms and grus," *arXiv preprint arXiv:1901.08987*, 2019.

[27] L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. S. Schoenholz, and J. Pennington, "Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks," *arXiv preprint arXiv:1806.05393*, 2018.

[28] G. Yang and S. Schoenholz, "Mean field residual networks: On the edge of chaos," in *Advances in neural information processing systems*, 2017, pp. 7103–7114.

[29] R. Karakida, S. Akaho, and S.-i. Amari, "Universal statistics of fisher information in deep neural networks: Mean field approach," *arXiv preprint arXiv:1806.01316*, 2018.

[30] J. Pennington, S. S. Schoenholz, and S. Ganguli, "The emergence of spectral universality in deep networks," *arXiv preprint arXiv:1802.09979*, 2018.

[31] C. K. Williams, "Computing with infinite networks," in *Advances in neural information processing systems*, 1997, pp. 295–301.

[32] A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani, "Gaussian process behaviour in wide deep neural networks," *arXiv preprint arXiv:1804.11271*, 2018.

[33] A. Garriga-Alonso, L. Aitchison, and C. E. Rasmussen, "Deep convolutional networks as shallow gaussian processes," *arXiv preprint arXiv:1808.05587*, 2018.

[34] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as gaussian processes," *arXiv preprint arXiv:1711.00165*, 2017.

[35] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, "The implicit bias of gradient descent on separable data," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 2822–2878, 2018.

[36] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro, "Characterizing implicit bias in terms of optimization geometry," *arXiv preprint arXiv:1802.08246*, 2018.

[37] S. Gunasekar, J. D. Lee, D. Soudry, and N. Srebro, "Implicit bias of gradient descent on linear convolutional networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 9482–9491.

[38] Z. Ji and M. Telgarsky, "Gradient descent aligns the layers of deep linear networks," *arXiv preprint arXiv:1810.02032*, 2018.

[39] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," *arXiv preprint arXiv:1811.04918*, 2018.

[40] L. Wu, C. Ma, and E. Weinan, "How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective," in *Advances in Neural Information Processing Systems*, 2018, pp. 8279–8288.

[41] B. Neyshabur, R. Tomioka, R. Salakhutdinov, and N. Srebro, "Geometry of optimization and implicit regularization in deep learning," *arXiv preprint arXiv:1705.03071*, 2017.

[42] Z. Zhu, J. Wu, B. Yu, L. Wu, and J. Ma, "The anisotropic noise in stochastic gradient descent: Its behavior of escaping from minima and regularization effects," in *International conference on machine learning*, 2019.

[43] R. Kleinberg, Y. Li, and Y. Yuan, "An alternative view: When does sgd escape local minima?" *arXiv preprint arXiv:1802.06175*, 2018.

[44] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, "Entropy-sgd: Biasing gradient descent into wide valleys," *arXiv preprint arXiv:1611.01838*, 2016.

[45] P. Chaudhari, A. Oberman, S. Osher, S. Soatto, and G. Carlier, "Deep relaxation: partial differential equations for optimizing deep neural networks," *Research in the Mathematical Sciences*, vol. 5, no. 3, p. 30, 2018.

[46] Y. W. Teh, A. H. Thiery, and S. J. Vollmer, "Consistency and fluctuations for stochastic gradient langevin dynamics," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 193–225, 2016.

[47] C. Li, C. Chen, D. Carlson, and L. Carin, "Preconditioned stochastic gradient langevin dynamics for deep neural networks," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[48] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 IEEE Information Theory Workshop (ITW)*. IEEE, 2015, pp. 1–5.

[49] A. Achille, M. Rovere, and S. Soatto, "Critical learning periods in deep networks," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=BkeStsCcKQ

[50] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 1947–1980, 2018.

[51] ——, "Where is the information in a deep neural network?" *CoRR*, vol. abs/1905.12213, 2019. [Online]. Available: http://arxiv.org/abs/1905.12213

[52] G. Valle-Perez, C. Q. Camargo, and A. A. Louis, "Deep learning generalizes because the parameter-function map is biased towards simple functions," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rye4g3AqFm

[53] S. Goldt and U. Seifert, "Stochastic thermodynamics of learning," *Physical review letters*, vol. 118, no. 1, p. 010601, 2017.

[54] ——, "Thermodynamic efficiency of learning a rule in neural networks," *New Journal of Physics*, vol. 19, no. 11, p. 113001, 2017.

[55] S. Yaida, "Fluctuation-dissipation relations for stochastic gradient descent," *arXiv preprint arXiv:1810.00004*, 2018.

[56] E. Weinan, "A proposal on machine learning via dynamical systems," *Communications in Mathematics and Statistics*, vol. 5, no. 1, pp. 1–11, 2017.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[58] Y. Lu, A. Zhong, Q. Li, and B. Dong, "Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations," *arXiv preprint arXiv:1710.10121*, 2017.

[59] S. Sonoda and N. Murata, "Transport analysis of infinitely deep neural network," *Journal of Machine Learning Research*, vol. 20, no. 2, pp. 1–52, 2019. [Online]. Available: http://jmlr.org/papers/v20/16-243.html

[60] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems*, 2018, pp. 6572–6583.

[61] R. T. Q. Chen and D. Duvenaud, "Neural networks with cheap differential operators," in *2019 ICML Workshop on Invertible Neural Nets and Normalizing Flows (INNF)*, 2019.

[62] B. Hu and L. Lessard, "Control interpretations for first-order optimization methods," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 3114–3119.

[63] E. Weinan, J. Han, and Q. Li, "A mean-field optimal control formulation of deep learning," *arXiv preprint arXiv:1807.01083*, 2018.

[64] Q. Li, L. Chen, C. Tai, and E. Weinan, "Maximum principle based algorithms for deep learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5998–6026, 2017.

[65] Q. Li and S. Hao, "An optimal control approach to deep learning and applications to discrete-weight neural networks," *arXiv preprint arXiv:1803.01299*, 2018.

[66] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Accelerating adversarial training via maximal principle," *arXiv preprint arXiv:1905.00877*, 2019.

[67] G. A. Pavliotis, *Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations*. Springer, 2014, vol. 60.

[68] U. Simsekli, L. Sagun, and M. Gurbuzbalaban, "A tail-index analysis of stochastic gradient noise in deep neural networks," *arXiv preprint arXiv:1901.06053*, 2019.

[69] Q. Li, C. Tai, and W. E, "Stochastic modified equations and adaptive stochastic gradient algorithms," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2101–2110.

[70] J. An, J. Lu, and L. Ying, "Stochastic modified equations for the asynchronous stochastic gradient descent," *arXiv preprint arXiv:1805.08244*, 2018.

[71] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[72] Y. Li and Y. Liang, "Learning overparameterized neural networks via stochastic gradient descent on structured data," in *Advances in Neural Information Processing Systems*, 2018, pp. 8157–8166.

[73] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in neural information processing systems*, 2018, pp. 8571–8580.

[74] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[75] A. Nøkland and L. H. Eidnes, "Training neural networks with local error signals," *arXiv preprint arXiv:1901.06656*, 2019.

[76] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," *arXiv preprint arXiv:1611.05397*, 2016.

[77] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor, "Learning end-to-end multimodal sensor policies for autonomous navigation," *arXiv preprint arXiv:1705.10422*, 2017.

[78] V. G. Boltyanskii, R. V. Gamkrelidze, and L. S. Pontryagin, "The theory of optimal processes. i. the maximum principle," TRW SPACE TECHNOLOGY LABS LOS ANGELES CALIF, Tech. Rep., 1960.

[79] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[80] R. E. Bellman and R. E. Kalaba, *Selected papers on mathematical trends in control theory*. Dover Publications, 1964.

[81] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[82] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems*. Addison-Wesley Reading, MA, 1994, vol. 3.

[83] G. M. Rotskoff and E. Vanden-Eijnden, "Trainability and accuracy of neural networks: An interacting particle system approach," *arXiv preprint arXiv:1805.00915v3*, 2019.

[84] B. Øksendal, "Stochastic differential equations," in *Stochastic differential equations*. Springer, 2003, pp. 65–84.

[85] S. L. Smith and Q. V. Le, "A bayesian perspective on generalization and stochastic gradient descent," *arXiv preprint arXiv:1710.06451*, 2017.

[86] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey, "Three factors influencing minima in sgd," *arXiv preprint arXiv:1711.04623*, 2017.

[87] E. Moulines and F. R. Bach, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in *Advances in Neural Information Processing Systems*, 2011, pp. 451–459.

[88] G. N. Milstein, *Numerical integration of stochastic differential equations*. Springer Science & Business Media, 1994, vol. 313.

[89] K. Itô, *On stochastic differential equations*. American Mathematical Soc., 1951, vol. 4.

[90] A. Kolmogoroff, "Über die analytischen methoden in der wahrscheinlichkeitsrechnung," *Mathematische Annalen*, vol. 104, no. 1, pp. 415–458, 1931.

[91] R. Jordan, D. Kinderlehrer, and F. Otto, "The variational formulation of the fokker–planck equation," *SIAM journal on mathematical analysis*, vol. 29, no. 1, pp. 1–17, 1998.

[92] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein, "Metastability in reversible diffusion processes i: Sharp asymptotics for capacities and exit times," *Journal of the European Mathematical Society*, vol. 6, no. 4, pp. 399–424, 2004.

[93] W. Hu, C. J. Li, L. Li, and J.-G. Liu, "On the diffusion approximation of nonconvex stochastic gradient descent," *arXiv preprint arXiv:1705.07562*, 2017.

[94] W. Xu, "Towards optimal one pass large scale learning with averaged stochastic gradient descent," *arXiv preprint arXiv:1107.2490*, 2011.

[95] S. P. Coraluppi and S. I. Marcus, "Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes," *Automatica*, vol. 35, no. 2, pp. 301–309, 1999.

[96] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.

[97] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 681–688.

[98] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[99] I. Goodfellow, "Defense against the dark arts: An overview of adversarial example security research and future research directions," *arXiv preprint arXiv:1806.04169*, 2018.

[100] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," in *Advances in Neural Information Processing Systems*, 2018, pp. 5014–5026.

[101] E. Wong and J. Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," *arXiv preprint arXiv:1711.00851*, 2017.

[102] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[103] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl^2: Fast reinforcement learning via slow reinforcement learning," *arXiv preprint arXiv:1611.02779*, 2016.

[104] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," *arXiv preprint arXiv:1801.08930*, 2018.

[105] C. Tao, S. Dai, L. Chen, K. Bai, J. Chen, C. Liu, R. Zhang, G. Bobashev, and L. C. Duke, "Variational annealing of gans: A langevin perspective," in *International Conference on Machine Learning*, 2019, pp. 6176–6185.

[106] G. Yang, J. Pennington, V. Rao, J. Sohl-Dickstein, and S. S. Schoenholz, "A mean field theory of batch normalization," *arXiv preprint arXiv:1902.08129*, 2019.