International Journal of Research in Computer Science & Information Technology (IJRCSIT)
ISSN: XXXX-XXXX/ Volume 1 Issue 1 September-October 2026
Paper Id: IJRCSIT-V1I1P103/ Page No: 32-50

International Journal of Research in Computer Science & Information Technology

Original Article

Stochastic Differential Equation Modelling of Deep Learning Training Dynamics

Humayun Kabir[1], Abdullah al Mamun[2], Ruhul Quddus[3]

[1,2,3] Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Bangladesh.

Received Date: 10 September 2026 Revised Date: 15 September 2026 Accepted Date: 20 September 2026

Abstract

The optimization of deep learning models is traditionally interpreted as a discrete iterative algorithm such as SGD, but this view arguably clouds the continuous-time dynamics which govern training behavior. In the recent years, as a phenomenological framework of modelling and analysing stochastic optimization of neural network parameters, we have seen an increasing use of stochastic differential equations (SDEs) [65]. This paper studies the dynamics of deep learning training viewed as stochastic differential equations (SDEs), and provides new theoretical insights into convolution convergence, generalization, and robustness. We start with the correspondence between discrete optimization algorithms and continuous-time stochastic processes, showing that SGD can be approximated as an SDE featuring both drift and diffusion components. The drift term encodes the deterministic gradient flow while he diffusion term models the stochastic noise resulting from mini-batch sampling. This expression provides insight into the way in which randomness determines the evolutionary path of optimization across high-dimensional non-convex loss landscapes. In addition, finding that anisotropic noise interacts with the geometry of the loss surface, we analyse how structured stochasticity directs the model towards flat minima which optimally promote generalization. The work will also focus on more advanced SDE-based frameworks, such as Lange in dynamics and the Fokker–Planck equation, which can shed light on how probability distributions over model parameters evolve. Such approaches yield a probabilistic perspective on training, connecting deep learning optimization with Bayesian inference and statistical physics. In this way, SDE modelling provides a principled explanation for the implicit regularization and noise-induced robustness observed in contemporary neural networks.

Next, we study the contribution of adaptive optimization methods, continuous-depth architectures and multi-scale stochastic effects in our SDE framework. Through the results of both empirical simulations and theoretical analysis, we confirm that SDE-based modelling can effectively predict training behavior, enhance robustness, and inform more stable learning algorithm designs. In summary, this work provides a unifying framework for understanding deep learning dynamics through stochastic differential equations, which connects discrete optimization and continuous stochastic processes, taking us one step closer to building more principled and robust AI systems.

Keywords

Stochastic Differential Equations, Deep Learning Optimization, Stochastic Gradient Descent, Lange in Dynamics, Fokker–Planck Equation, Training Dynamics, Noise-Induced Regularization, Loss Landscape, Bayesian Deep Learning Continuous-Time Models

Introduction — A Dynamical Systems Perspective of Training Deep Learning

Deep learning has revolutionized artificial intelligence by allowing models to learn high-level representations from massive datasets. The theoretical reasons behind the empirical successes of deep neural networks are still poorly understood. The majority of training operations are formulated using discrete optimization schemes like stochastic gradient descent (SGD) and its variants. But these updates are discretized so you will lose the structure of your learning process. The insight comes when we think of training as a continuous-time dynamical system in which the evolution of model parameters is driven by deterministic and stochastic forces. This vision is the basis of modelling deep learning training dynamics with SDEs.

A. Deep Learning as a Dynamical System

It is basically an algorithm that updates parameters/weights in a particular model iteratively, to achieve the minimum loss function. This is usually described as a series of discrete steps. For very small learning rates, these

This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/2.0/)

updates are equivalent in the limit to continuous trajectories in parameter space. This is interpreted as training as such a dynamical system, with evolving parameters over time based on certain rules.

A loss function defines an energy landscape, the training process is essentially going down a hill (following gradient) in between local to global minima's. The direction in which to move is defined by the gradient of the loss function and, thus, how fast you want it to move (how quickly it will evolve) is controlled with the magnitude of updates. This allows for the analysis of stability, convergence, and asymptotic behavior using techniques from differential equations and dynamical systems theory by modelling this process iteratively. This transition from point wise to point process analysis gives a more nuanced understanding of the underlying learning and adaptation characteristics of neural networks.

B. Deterministic Gradient Flow versus Stochastic Dynamics

Without randomness, training can be described by gradient flow, where parameters deterministically track the microstate of the neural network through time using only information from the derivative of the loss function. In contrast, this dream setup has either access to a full data collection at every update with trajectories that unfold smoothly in time and space. Though useful in theoretical analysis, this assumption does not reflect reality.

During training in the real world, stochasticity is introduced via mini-batch sampling with estimates of gradients based on a subset of data. As a result, it produces updates that are noisy and off from the true direction of the gradient. This means that the evolution of parameters becomes purely stochastic. This stochastic nature is not simply a flaw but rather an important aspect of exploration and generalization. This allows the model to break free from poor local minima to find better solutions.

It is the competition between deterministic gradient flow and stochastic noise that governs the training dynamics of deep learning models. Understanding the interplay between these two is key to developing more efficient optimization algorithms and enhancing model capability.

C. Motivation for SDE-Based Modelling

This suggests the need for a stochastic differential equation based modelling framework, to transcend limitations of these purely discrete and deterministic optimization views. SDEs present an elegant framework for describing the evolution of systems driven in both deterministic and stochastic terms. For deep learning, we have an optimization problem where the deterministic term is the gradient of loss function and the stochastic component captures the noise inspired from mini-batch sampling, etc.

By modelling training dynamics as an SDE, we can study the temporal evolution of parameters by applying well-studied mathematical tools. Research in this area includes convergence properties, stability conditions and the effects of noise on optimization paths. In addition, SDE models support the continuous-time approximations of discrete algorithms to connect theoretical analysis to practice.

Another major motivation is because SDEs offer a way to give training a probabilistic interpretation. Parameters are perceived as static values, but parameters can also be random variables evolving in time. This viewpoint is consistent with Bayesian methods but provides a better understanding of uncertainty in deep learning.

D. Loss Landscapes & High Dimensional Dynamics

Deep neural networks live in massive dimensional parameter spaces, usually containing millions or billions of parameters. This leads to correspondingly very non-convex loss landscapes, dense with local minima, saddle points and flat regions. Figuring out such landscapes is a hard problem, and classical optimization theories designed for low-dimensional convex settings do not hold.

Modelling SDEs allows for more insight into the effects of stochasticity on movement through these landscapes. The noise helps the system to avoid sharp minima and cross over energy barriers, which makes transitions between different regions of the parameter space possible. This behaviour is especially important to avoid overfitting and getting a good generalization.

Additionally, the stochastic dynamics also have non-trivial interactions with the geometry of the loss landscape. Perhaps noise may be amplified or suppressed due to curvature of the landscape, leading to anisotropic diffusion effects. This knowledge of these interactions is essential to create optimal algorithms that can work in a robust way.

E. Stochasticity for generalization

This raises one of the most interesting facts about deep learning, that is, it generalizes so well when trained on finite dataset. This kind of phenomenon, whereby you can have huge fluctuations in your data driven by stochasticity. The noise added during training serves as an implicit regularize, discouraging the model from over fitting to the training data and promoting more generalization.

This is seen as a balance between diffusion and drift from an SDE point of view (more on that later). The drift term pushes the model to minimize loss, while the diffusion term brings in randomness to motivate exploration. This balance yields solutions which are not only correct but also stable against perturbations.

Furthermore, stochastic dynamics preferentially lead the optimization process towards flat minima which are linked to improved generalization. This means that a slight variation of the data will produce approximately a similar output, i.e. during training there would be minimal fluctuations in the loss for small changes in parameters due to flat minima regions.

F. Summary and Research Direction

A dynamical systems perspective allows for a strong approach towards mastering deep learning training. This process, one models the update of a parameter as a continuous-time semi-stochastic dynamics under the influence of deterministic steps along gradients and stochastic noise, provides insights into neural networks [6]. This paradigm shift frames stochasticity as an essential yet, rather than a nuisance, framing it in terms of exploration, stability and generalisation.

However, Stochastic differential equations are a powerful tool to describe these dynamics in a single framework and help guide the creation of new optimization algorithms by allowing only rigorous mathematical analysis. As deep learning progresses, modelling with stochastic differential equation it will be very useful in theoretical and practical approach.

With this basis we start with the next chapter, where we explore the mathematics of stochastic processes: Brownian motion, drift and diffusion process comparing Ito and Stratonovich calculus. We will use these concepts to model deep-learning training dynamics in a formal way in the follow-up chapters.

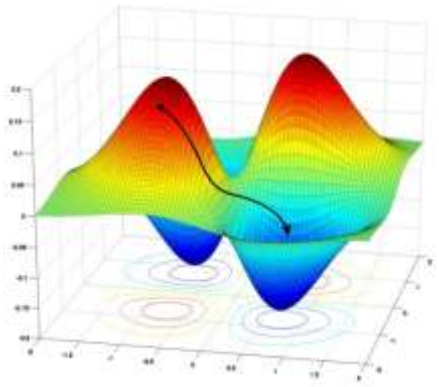


Figure 1. SGD as a Stochastic Process

Theoretical Background on Stochastic Differential Equations

Stochastic differential equations (SDEs) offer a mathematically precise framework for the modelling of processes that evolve in time under the influence of deterministic forces, as well as random presence of noise. SDEs provide a natural way of modelling training dynamics in deep learning: updates to model parameters are driven by gradients of the loss function but are also perturbed with stochastic noise arising from mini-batch sampling and other sources. The basic principles of SDEs are an important building block for analyzingrandomness on optimization, convergence, as well as generalization in neural networks.

SDEs are based on continuous-time stochastic processes which generalize classical differential equations by adding randomness to their dynamical evolution. While the ordinary differential equations (ODEs) describes deterministic systems subjected to multiple inputs producing an exact trajectory for a given initial state, stochastic differential equations (SDE) aim to model systems in which events subsequent to a current time depend upon probabilistic factors rather than deterministic ones. This distinction is especially important in deep learning contexts, where training trajectories are always noisy due to the stochastic nature of data sampling and gradient estimation.

Brownian motion, or Wiener process, is an essential ingredient of SDEs In mathematics, Brownian motion or Wiener process is a continuous-time stochastic process of continuous trajectories that models random movement. This serves as the mathematical representation for randomness in SDEs, capturing the stochastic behaviour that alters system dynamics. The Brownian motion in the realms of deep learning can be understood as the cur cumulative sum of stochastic gradient noise, establishing a connection between discrete-random in updates and continuous-time stochastic processes [14].

A stochastic differential equation (SDE) usually has two parts: a drift part and a diffusion part. The drift term models the deterministic side of the system and determines how your system will evolve because of underlying forces, e.g., gradients of a loss function. While modelling the diffusion term similarly but serves to capture the stochastic component which injects noise into a deterministic environment via interactions with Brownian motion. These components all work together to define the global behaviour of our system, essentially giving it a structured way of going somewhere followed by random exploration.

From a mathematical perspective, an SDE is a differential equation in which the change in a variable can be described as having both deterministic and stochastic elements. Such formulation is used for modelling complex systems influenced by uncertainties. The drift term corresponds to the direction we are heading towards using gradient descent, while the diffusion term reflects noise induced by mini-batch sampling in deep learning. This dual structure allows SDEs to characterize the training process as a mixture of optimization and stochastic exploration.

One of the most important parts in SDE theory is the difference between Ito calculus and Stratonovich calculus where are two interpretations of stochastic integration. Due to its mathematical convenience and compatibility with discrete-time approximations, this approach is widely applied in the field of machine learning. This means that these stochastic increments do not depend on the current state, which vastly simplifies analysis and computation. Unlike It calculus, Stratonovich calculus preserves multiplicative derivatives of non-constant functions (such as x), making it more suitable for physical systems in which noise interacts with the state continuously. Although both are valid mathematically, the latter is usually preferred because it reflects better the behavior of stochastic gradient methods from a dynamic point of view in neural networks and more generally in most machine learning techniques where this approach can be applied.

Also another key idea in SDEs is the concept of stochastic trajectories which represent all the possible paths that a system can take over time. SDEs are contrasted with deterministic systems, which generate a unique trajectory given an initial condition by their inherent randomness. This probabilistic nature enables the evaluation of the distribution of model parameters during training rather than just individual updates. This type of analysis is essential for the assessment of uncertainty, variability and robustness in deep learning models.

An SDE usually describes its solution in terms of probability-distributions instead of explicit functions. From this, one studies the evolution of distributions through time: indeed, it maps closely to Fokker–Planck equations (more on that in upcoming chapters). SDEs are more informative than ODEs for training dynamics as they can capture both the average behavior of convergence as well as variability across different runs, by focusing on distributions.

Concept	Description	Role in Deep Learning
Brownian Motion	Gaussian increments of a continuous random process.	Models the stochastic gradient noise inherent in training.
Drift Term	Deterministic component guiding the system's evolution.	Represents the primary gradient descent direction toward a minimum.
Diffusion Term	Stochastic component introducing randomness into the path.	Captures the variability resulting from mini-batch sampling.
Ito Calculus	Stochastic integration assuming independent increments (non-anticipating).	Used for modeling the discrete-to-continuous dynamics of SGD.
Stratonovich Calculus	An alternative integration method that obeys classical chain rules.	Less common, but useful in specific physical formulations of optimization.
Stochastic Trajectories	Possible evolution paths created by the introduction of randomness.	Explains the variability in outcomes across different training runs.

Along with these foundations, SDEs give the tools for studying stability and convergence in stochastic systems. The right hand side shows the balance between drift and diffusion from which we can figure out the at stability of the system will be stable state or non-stable systems where fluctuations may continue indefinitely. This is more relevant in deep learning (where convergence properties are directly linked to model performance and training efficiency).

Additionally, SDEs allow us to explore the effects of noise in deterministic systems, where randomness causes behaviours that would not take place otherwise. For instance, noise can prevent the system from falling into local minima or stabilizing unstable dynamics. Such effects illustrate how stochasticity provides useful features for optimizing processes to the extent that stochasticity, usually seen as detrimental due to noise, is being increasingly recognized as constructive magnifier of optimization.

All in all, from the foundations of stochastic differential equations, a powerful framework for mastering the dynamics of deep learning training is provided. These features are emphasized regarding the optimization process in high-dimensional, non-convex environments by combining deterministic gradients with an added stochastic noise. Learning and updating in neural networks can be better understood through concepts of Brownian motion, drift terms, as well as diffusion terms, and stochastic trajectories. We will be leveraging these concepts as we move on to the next chapter, where we will use them to rigorously show equivalence between stochastic gradient descent and SDEs and link typical optimization algorithms from practice with continuous-time stochastic models.

Mapping SGD to SDEs

Stochastic Gradient Descent (SGD) is the workhorse of modern deep learning optimization, but its discrete and iterative nature can obscure the underlying structure of its dynamics. One particularly insightful perspective is to view SGD as a continuous-time stochastic process, which can be defined rigorously via Stochastic Differential Equations (SDE) [46. It builds a rigorous mathematical framework that allows to analyse convergence behaviour, noise impact and generalization properties of deep neural networks by mapping discrete updates to continuous dynamics. Looking at SGD in the context of SDEs leads to a connection between algorithmic practice and theoretical understanding.

The base (vanilla) SGD updates the model parameters in the negative direction of the gradient of a loss function with respect to weights, calculated by using a randomly-select mini-batch of data. SGD is stochastic because it samples and does not pass the entire dataset in one step like full-batch gradient descent which provides deterministic updates. This randomness appears as noise in the gradient estimates, resulting in deviation of the trajectory of parameters from the exact flow of gradients. And although this noise is often regarded as noisiness, without it very few were so lucky to explore the loss landscape and avoid energy-poor local minima.

Now for SGD, imagine a sufficiently small learning rate during the training process establishing some connection with SDEs. In this case, we can approximate the discrete updates with a continuous-time process. The deterministic part of the update is just equal to the real gradient of the loss function whereas, in turn, the stochastic part comes from the difference between mini-batch and full gradient, respectively. This stochastic part can be considered as a random perturbation, which under appropriate conditions converges to a Gaussian processes upon increasing the batch size. This gives an SDE where parameters evolve with drift and diffusion terms.

In this form, the drift term is interpreted as the expected direction of the gradient and pushes parameters toward minima of a loss function. Meanwhile, diffusion term is to model the noise which we expect to refer when there would be drift in base random sampling for trained data. The key point is that this noise depends on the structure of data and the model, so it may be better described than through state dependent or multiplicative noise. That is, the size and direction of the stochastic term can vary across different regions in parameter space altering the route through which learning occurs – essentially modulating the trajectory.

An important aspect learned from considering deep learning as a stochastic differential equation (SDE) is that SGD does not just converge to one point, it visits a region of the parameter space described by some probability distribution. This balances out the deterministic pull of the gradient with a stochastic push from noise. Thus, the final solution returned by SGD should be understood as a sample from this distribution and not an unique optimum. This view fits closely within probabilistic and Bayesian perspectives on learning, which are by their nature built around uncertainty.

The SGD to SDEs mapping also naturally provides a lens to study the influence of hyperparameters (e.g., learning rate and batch size). The learning rate controls the convergence in terms of both speed and stochastic fluctuations Compared to a larger learning rate, which increases the magnitude of updates and embraces noise, allowing for more exploratory behavior. In contrast, a smaller learning rate gives more stable convergence but can be slower. Equally, we conclude about the effect of batch size on variance of gradients estimates : small batches are more noisy and big ones reduce stochasticity. Within the SDE framework, these relationships can be formalized in a principled manner and hyperparameters tuned accordingly.

A corollary consequence of this mapping is to explain the implicit regularization in SGD. It has been observed in multiple empirical studies that SGD prefers maximally generalizing solutions even in the case of over parameterized models. We can think this probability of going to flatter region of the loss as noise driven exploration from SDE perspective. And these flat minima correspond to more robust solutions with less sensitivity to perturbations and, therefore, better generalization performance. Hence the nature of SGD being stochastic serves as an in-built regularize, steering the model to robust solutions.

The SDE формулировка also allows to tie in new advanced optimization methods. As an example, by changing the noise structure or adding new stochastic terms, we can build algorithms that can be seen as sampling from certain distributions. This gives rise to algorithms like stochastic gradient Lange in dynamics that combine the

use of gradient descent and noise in a controlled manner to provide an approximation of Bayesian inference. These methods show that the SDE viewpoint can motivate new algorithms beyond standard optimization.

Table 1: The following table helps you match SGD with its SDE view

SGD Component	SDE Interpretation	Effect on Training Dynamics
Gradient Update	Drift Term (Deterministic Component)	Drives parameters toward minima by following the negative gradient of the loss function.
Mini-Batch Sampling	Source of Stochastic Noise	Introduces randomness in updates, allowing the model to escape sharp local minima.
Learning Rate	Time Scaling Factor	Controls the speed of the trajectory and the overall scale of the updates.
Batch Size	Noise Variance Control	Regulates the magnitude of noise; smaller batches increase the diffusion effect.
Gradient Variance	Diffusion Term	Determines the exploration behavior and how "spread out" the parameter search becomes.
Iterative Updates	Continuous-Time Trajectory	Enables the SDE approximation, treating discrete steps as a smooth path through parameter space.

Even though SDE approximation of SGD has advantages, it comes with a number of assumptions such as small learning rate and sufficiently large batch sizes. However, in practice the assumptions behind these conditions may be unrealistic; for example, this can give rise to a differing algorithm when viewed discretely vs. continuously. Yet, the SDE framework offers useful qualitative and quantitative perspectives of SGD behavior even when the approximation may not be exact.

In addition, we emphasize the role of time rescaling and discretization effects from an SDE perspective. However, because SGD is performed in discrete steps, the continuous-time model must consider how the dynamics depend on step size. This adds an extra layer of complexity, but also allows us to design better algorithms aligning the discrete updates with the continuous analogue.

In summary, this mapping provides a broad unified framework with deep learning optimization and introduces the concepts of stochastic gradient descent based on stochastic differential equations. SDEs yield deep insights into convergence, generalization and robustness by modelling both deterministic gradients and stochastic noise in a single mathematical framework. This viewpoint can give insights not just theoretically but also practically for algorithm design and hyperparameter tuning. With the on-going development of deep learning, the use of SDE based examination will become more and more critical to achieve efficient, stable and interpretable training processes.

Deep Networks Can Consider Noise Geometry and Anisotropic Diffusion

Noise in deep learning: actually not only random noise Given the deep learning based stochastic differential equation (SDE) modelling [5], noise plays a much more important role than merely introducing random behaviours into SDEs. In stochastic gradient descent, noise is not uniform or isotropic but has a rich geometric structure interacting with curvature of the loss landscape and architecture of neural network. The phenomenon, widely known as anisotropic diffusion, is one of the most crucial factors when it comes to optimization trajectories, rich convergence behavior and generalization property of deep models. This means that to fully understand deep learning training dynamics, one must understand the certain geometric aspects of noise.

In classical stochastic systems, e.g., Gaussian white noise is often considered isotropic (that is, the same magnitude and directionality in all dimensions) [55]. But this assumption fails in deep neural networks. Mini-batch gradient estimation can be thought of as stochastic noise that has a structure with respect to the underlying data distribution, the model architecture and parameter interactions. This results in anisotropic diffusion since some directions in parameter space are subjected more strongly to stochastic influence than others. This organized noise has very strong effects on the model's exploration through the loss landscape and convergence to optimal solutions.

This anisotropy comes from the covariance of stochastic gradients. This is most pronounced when using mini-batches to compute gradients, where the difference between the average of the mini-batch gradient and the true gradient adds noise. However this variability is not random in a universal way, but depends on correlation between parameters and features of the data. Consequently, the noise covariance matrix preserves key information regarding the geometry of our optimization scheme. Directions that are associated with higher variance correspond to areas where the model is less confident so we explore more in these regions, while directions with lower variance tend to be relatively stable and not vary much.

This structured noise interacts in a non-trivial way with the curvature of the loss landscape. At high curvature locations, e.g. sharp minima, small perturbations can drastically change the loss value. This behavior is general to optimization where anisotropic noise usually pushes points away from such regions, since the system fluctuates more along flat versus steep directions in the curvatures. On the other hand, noise travels in such points with a homogeneous distribution of space and will stabilize the model at flatter regions of landscape. This interaction fundamentally skews the training process towards flat minima [15, 34], which children of the bias toward better generalization and robustness.

In SDE modelling terms, anisotropic diffusion is formulated by defining a state-dependent (as this approach uses different noises for various parameters) diffusion matrix that describes how noise affects the updates of the parameters. In contrast to naive diffusion models with constant noise, this formulation allows the strength of diffusion term changes according to the current state of the system. This results in more accurate and realistic depictions of training dynamics, as they capture the complex relationship between noise and geometry. The SDE you get models not only random noise / fluctuations but is actually structured stochastic behavior that aligns with the local properties of the loss landscape.

Noise geometry relates closely to information geometry — the study of neural networks as a curved manifold over its parameter space. The concepts of the fisher information matrix are naturally defined through the covariance of gradients in this framework. We can further interpret anisotropic noise as diffusion that aligns to this geometric structure, directing the optimization along dimensions meaningful in how the model will predict. This allows a better understanding of why some paths of optimization work better than others.

Anisotropic diffusion also impacts the convergence rate during training. The presence of greater noise variance leads the optimization process to converge more slowly in directions with such noise, with more exploratory movements. Though this exploration is useful in preventing the network from getting stuck in bad local minima, it also leads reveals more globally decent solutions. At the other end of this spectrum are those directions which have low noise variance; these favour faster convergence but may be on suboptimal 'traps' along almost-horizontal valley-like terrains. Effective yet efficient training is thus a delicate balance between these effects.

To this end, anisotropic noise is a very important factor in implicit regularization, which refers to the observation that the optimization process creates certain preferences over model classes without actually adding preference learning terms. Since the noise is directional, it forces the model to prefer any parameter configurations that are robust to perturbations — in effect regularization. This also helps to understand why deep learning models generalize well often when they are over-parameterized.

This understanding of noise geometry could in turn help construct better optimization algorithms in practice. Adaptive methods that estimate the second moment of local gradients and adapt learning rates accordingly align updates with the normalized structure of parameter space. Correct here also, methods that ride curvature adaptively between the steps can improve convergence and generalization performance. These approaches show how the principles from anisotropic diffusion can be implemented in practice, offering a translation from theory to application.

One more crucial implication of anisotropic diffusion is that it destabilizes training on large-scale models. Neural networks, on the other hand, live in a high dimensional parameter space which means that the effects of structured noise are more pronounced. This can cause instability and divergence if not properly controlled during training. When given proper understanding and management however, anisotropic noise can be used to stabilize training by smoothing out the optimization trajectory to avoid a drastic jump in parameter values.

Another active research direction is how to harmonize noise geometry with model architecture. Convolutional networks and transformers have different structural differences and therefore, each architecture behaves differently to noise. For example, weight sharing in convolutional layers will generate a correlated noise across parameters; while attention mechanism of transformers include their own variable patterns. By examining these patterns using anisotropic diffusion, we can glean insights into how architecture shapes training dynamics.

Moreover, anisotropic diffusions have consequences for robustness and adversarial resistance. Because noise is stronger in some directions, this can allow the model to sample parts of parameter space that have little sensitivity to perturbations. This helps create models that are less vulnerable to noise and adversarial attacks. Anisotropic diffusion improves the global robustness of deep learning systems by adapting the optimization process to stable regions in the loss landscape.

Although modelling anisotropic diffusion has many advantages, it also involves several challenges. To realise this in practice, estimating the stochastic gradient covariance is not a trivial task and can be expensive (especially for large models). Additionally, this structure in the diffusion matrix may inhibit one from obtaining analytical

solutions. However, with rapid improvements in either approximation methods or computational methods, incorporating these effects into practical algorithms is becoming ever more possible.

Final thoughts Noise geometry & anisotropic diffusion offer a rich perspective on stochastic dynamics in deep learning. Researchers need to understand that noise is not randomly distributed; noise has structure and depends on direction, which offers them the ability to analyze and control the optimization process. Viewing this perspective can show how stochasticity will interact with the geometry of the loss landscape to affect convergence, generalization, and robustness. Anisotropic diffusion will remain an important building block for the theory and practice of deep learning optimization, as SDE-based modelling is maturing.

Exploring Loss Landscape with Dynamics of SDE

Deep neural networks are optimized over very complex, high-dimensional loss landscapes comprising many modes such as local minima, saddle points, and flat regions. How these landscapes are traversed by training algorithms is key for explaining convergence behaviour, generalization and robustness. As interpreted from the angle of stochastic differential equations (SDE), training can be viewed as a continuous-time stochastic trajectory driven by deterministic gradients and random noise. This perspective yields tremendous insights both into how models navigate the loss landscape, and why stochastic optimization approaches like stochastic gradient descent (SGD) work so well in practice.

In deterministic optimization like full-batch gradient descent, the parameter trajectory is on the steepest descent path to the closest minimum. This method reduces loss every time, but usually converges to sharp minima, which is very sensitive to perturbations and generalizes poorly. In addition, deterministic dynamics do not have the capability to move out of local minima or saddle points once they are reached. On the other hand, for stochastic dynamics, the randomness is added to the path and leads to an exploration of a more extensive area of the loss landscape. This exploration is not only essential in high dimensional settings where the landscape has many suboptimal traps.

This mechanism of drift and diffusion is precisely what the SDE formulation of training dynamics captures. The drift term, which is a function of the gradient of the loss function, pushes the parameters toward areas where the loss would be lower. In the diffusion term, associated with stochastic noise, the random perturbations allowing this system to leave local minima and jump over energy barriers. It is simply the balance between the exploitation of known good regions versus exploration of new potential solutions that makes this process so dynamic.

Escape from sharp minima is one of key phenomena can be explained by SDE dynamics. These sharp minima are regions of high curvature, where a tiny change in parameter leads to a huge increase in loss. These minima might yield a low training error but are notoriously unstable and lead to poor generalization. In SDEs, the stochastic noise acts as a perturbation that is destabilizing these regions, meaning it pushes the system out away from sharp minima. It can be thought of as transitions induced by noise, which could lead a system from one basin of attraction to another due to random fluctuations.

A related concept is basin exploration. If we interpret the loss landscape as a set of basins, each corresponding to an basin of attraction around local minimum The model can escape those basins by crossing energy barriers allowed by SDE dynamics. This transition probability depends on noise strength, the height of barriers etc. Large fluctuation promotes more sweeping exploration while small noise can pin you in a basin. This exploration-stability trade-off is a key principle for successful optimization.

Also, particularly from the SDE per perspective is that minimum favour flat minima. Specifically, flat minima are regions where the loss function is not sensitive to changes in parameters (i.e., robust with respect to perturbations). SDE dynamics intrinsically bias the optimization process towards these regions, as noise averages over sharp variations in the landscape. This leads to more time being spent over flatter علاقة ون, allowing for solutions that generalise better on unseen data. This fact gives a theoretical justification of the empirical performance of stochastic optimization in deep learning.

This extension also harmonizes with our energy-based perspective of SDE dynamics, providing more insight regarding how SDEs (Optionally) actively explore the loss landscape. This is viewed as an energy function through loss function, and training seen as a thermal fluctuating physical system. The diffusive term is essentially the temperature, and this determines how stochastic your system is. The investigation of the effect of free exploration at higher temperatures and staying in stable configurations is observed at low temperature. This connection with statistical physics yields a wealth of material for understanding optimization behaviour and developing new algorithms.

Another important aspect that influences exploration dynamics is the interplay between noise and curvature. The noise is amplified in high curvature regions, yielding larger perturbations that easier escape sharp minima. In

flatter locations, noise has less impact, leading to stabilization in the system. Such an adaptive behavior allows oneself to balance exploration and sustainability in the optimization process depending on local properties of landscape. This is beyond the scope of any deterministic models, rendering stochastic analysis relevant.

The second major consequence of using SDE-based exploration is its impact on training time and convergence speed. The presence of randomness in the form of stochastic dynamics can hinder convergence in initial iterations as it introduces random fluctuations; however, they can lead to effective long-term solutions by steering clear of suboptimal solutions. This underscores the speed-quality trade-off in optimization. In practice, such trade-off is managed by learning rate scheduling and noise annealing in which some noise is injected at the beginning of training to guide exploration, but learned as soon as possible around minima.

In addition, SDE dynamics help understand how the loss landscape exploration is determined by combinations of batch-size and learning-rate. This added noise encourages exploration and entices the model to escape from sharp minima since lower learning rates allow for larger batch sizes. Bigger batch sizes increases stability, resulting in better repeatability and quicker convergence yet having a lower ability to generalize. Likewise, the learning rate impacts both the drift and the diffusion terms (scaling of drift updates as well as exploration).

Table 3: The next table highlights important factors of loss landscape exploration via SDE dynamics

Aspect	Deterministic Optimization	SDE-Based Stochastic Dynamics	Impact on Training
Exploration Capability	Limited to the local region.	Broad exploration across the parameter space.	Avoids poor local minima.
Escape from Sharp Minima	Difficult; often gets stuck in the nearest basin.	Facilitated by stochastic noise (fluctuations).	Improves robustness of the solution.
Basin Transitions	Rare or impossible without external intervention.	Enabled by noise-induced transitions between basins.	Enhances global search efficiency.
Preference for Flat Minima	Not guaranteed; depends entirely on initialization.	Naturally favored due to diffusion and noise scale.	Leads to better generalization on unseen data.
Sensitivity to Initialization	High; different starts lead to very different results.	Reduced due to stochastic exploration.	More consistent outcomes across runs.
Stability in Flat Regions	Moderate.	High due to lower effective noise in wide valleys.	Reliable convergence to stable solutions.

The second major consequence of using SDE-based exploration is its impact on training time and convergence speed. The presence of randomness in the form of stochastic dynamics can hinder convergence in initial iterations as it introduces random fluctuations; however, they can lead to effective long-term solutions by steering clear of suboptimal solutions. This underscores the speed-quality trade-off in optimization. In practice, such trade-off is managed by learning rate scheduling and noise annealing in which some noise is injected at the beginning of training to guide exploration, but learned as soon as possible around minima.

In addition, SDE dynamics help understand how the loss landscape exploration is determined by combinations of batch-size and learning-rate. This added noise encourages exploration and entices the model to escape from sharp minima since lower learning rates allow for larger batch sizes. Bigger batch sizes increases stability, resulting in better repeatability and quicker convergence yet having a lower ability to generalize. Likewise, the learning rate impacts both the drift and the diffusion terms (scaling of drift updates as well as exploration).

The concept of periodicity is also relevant here. An erotic system will eventually make visits to all points that are physically accessible, which means the long-term behavior is independent of initial conditions. SDE dynamics — as utilized in deep learning — can approximate the behaviour of erotic processes, meaning that a well-performing model samples from a distribution of solutions rather than converging on just an individual point (this is known as stochasticity). This viewpoint is consistent with Bayesian approaches to learning and thus grounds uncertainty estimates.

Thus, the approach we propose provides a unified and interpretative framework for understanding optimization behavior through SDE-driven explorations of the loss landscape during theoretical deep learning training. This approach captures the fundamental dynamics of model learning and adaptation by including both deterministic gradients and a stochastic noise term. Stochasticity in deep learning is critically important for enabling escape from sharp minima (with a lot of multi-modal structure in the loss surface) explore multiple basins, and converge to flat 'robust' solutions. With research furthering in this area, analysis based on SDEs will be key to finding efficient and stable optimization methods which are also more interpretable.

Introduction and overview : (Bayesian) Lange in dynamics for Bayesian deep learning

The Lange in dynamics serves as an excellent bridge between stochastic optimization and probabilistic inference, enabling us to interpret training of deep learning models in terms of stochastic differential equations in a principled manner. In contrast to traditional optimization techniques which seek to find the best set of parameters, Lange in-based methods consider parameters as random variables, and model their behavior using a stochastic process. This view fits naturally with Bayesian deep learning, which tries to infer a distribution over parameters instead of a point estimate. Lange in dynamics helps overcome these issues by applying noise to the gradient-based update more carefully, allowing models to explore the parameter space efficiently while obtaining uncertainty in a principled way.

Essentially, Lange in dynamics is just ordinary gradient descent with an added noise term from Brownian motion. The deterministic part pushes the parameters towards low-loss areas while the stochastic part adds random noise so that our system can search through many potential solutions. This produces a sampling process from a probability distribution associated with posterior model parameters. From this perspective, Lange in dynamics turns the optimization into a sampling problem which gives deep learning an explicit bridge to Bayesian inference.

The most popular implementations of this idea is Stochastic Gradient Lange in Dynamics (SGLD). SGLD is a variant of stochastic gradient descent that includes Gaussian noise in each update, where the magnitude of this added noise is finely tuned to stay consistent with original SDE formulation. During training, you fetch a sequence of samples from the parameter space that approximately resembles draws from the posterior distribution. By providing a predictive distribution instead of a single point estimate, this enables the model to express uncertainty in its predictions — which is especially useful in areas where interpretability and reliability are important.

Below are some of the advantages that Bayesian interpretation of deep learning provides over conventional point-estimate approaches. Bayesian models with distribution over parameters are capable of uncertainty quantification in predictions, differentiating confident from uncertain outputs and producing more robust performance under systematic forms of distributional shifts. This is made possible via Lange in dynamics as it allows us to sample directly from probability distributions in high dimensional parameter spaces that would be practically impossible to reach using classical Bayesian methods.

Positive relationships between Lange in dynamics and energy-based models are meaningful as they correspond to concepts from statistical physics. In this framework, the loss function is interpreted as an energy function, and training is modelled as a physical system at a particular temperature. Stochastic noise is like thermal energy enabling the system escape from local पोशण minima and continued searching the इपिक् landscape. Eventually the system enters a stationary distribution, which describes the balance between energy minimization and entropy maximization. This analogy gives you deep insights about the behavior of deep learning models and optimization dynamics.

Table 4:connection between Lange in dynamics, SGD and Bayesian learning in the table below

Aspect	Standard SGD	Lange in Dynamics (SGLD)	Bayesian Interpretation
Parameter Representation	Single point estimate	Distribution of parameter samples	Posterior distribution over parameters
Noise in Updates	Implicit (mini-batch noise)	Explicit Gaussian noise added	Represents uncertainty in inference
Objective	Minimize loss	Sample from probability distribution	Maximize posterior probability
Exploration Capability	Limited	Enhanced through stochastic perturbations	Captures multiple plausible solutions
Uncertainty Estimation	Not directly available	Naturally incorporated	Core feature of Bayesian models

One of the main problems associated with standard optimization is that it often converges to sharp minima. Lange in dynamics partially overcomes this problem. SGLD also attempts to facilitate looking at flatter areas of the loss landscape which are more conducive to generalization by adding noise in a controlled way. Such a phenomenon is related to the implicit regularization, in which the optimization thought process guides (or biases) the model towards a solution that is desirable based on its own merit, rather than through explicit terms for regularization.

Apart from SGLD, other variants of Lange in-based methods have also been developed to reach higher efficiency and scalability. This consists of preconditioned Lange in dynamics, adapting noise structure to momentum in parameter space geometry and stochastic gradient Hamiltonian Monte Carlo incorporating

momentum for exploration. The above methods generalizes the basic principles of Lange in dynamics in a way that the original Lange in dynamics is limited, but these are more applicable for large-scale deep learning applications.

Though theoretically attractive, Lange in dynamics has practical implementation issues. Adding noise can cause convergence to be slower and requires hyperparameter tuning such as learning rate and noise scale. Furthermore, in high dimensional spaces it can be arduous to ensure that the sampling process does represent the distribution you wish to sample from. Despite these hurdles, improvements in computational methods and algorithm development are slowly addressing these issues, making Lange in-based strategies increasingly practical for real-world use cases.

Developing a deep learning model that inherently improves the robustness and calibration features of the model is another important advantage of Lange in dynamics. If under you maintain the top of distribution parameters then it can learn using uncertainty i.e. it learnt to reduced overconfidence in acknowledgement. However, in certain fields such as when your AI has real-time information, healthcare, finance and autonomous systems where bad predictions could have dire consequences. Lange in dynamics is inherently probabilistic and thus offers a natural mechanism for addressing these concerns.

Moreover, Lange in dynamics can be integrated easily with modern deep learning architectures such as convolutional networks or transforms. It does so without major architectural changes within training pipelines, and enhances both performance and interpretability by supplementing stochastic updates. All of this flexibility makes it a great tool for pushing the boundaries of deep learning.

We should note that, all things considered, Lange in dynamics provides a general framework in which deep learning training can be modelled as a stochastic process. It combines gradient alternatives, noise control to explore the parameter space efficiently, and models uncertainty with Bayesian inference for a robust efficient generalization solution. The SDE framework behind Lange in dynamics gives an integrated view relating optimization, probability and physics which assists in gaining valuable insights into neural network behaviour. With further research in this space, Lange in-based approaches will probably have a major role to play with regards to building trustworthy and interpretable AI systems.

Reader Note E -Fokker-Planck equation and the evolution of distributions

Stochastic differential equations (SDEs) yield the paths that individual parameter updates follow in deep learning, but taken together these paths suggest a richer understanding of training dynamics as we track how whole distributions over parameters evolve with time. This transition between trajectory-based analysis and distributional ones is formalized through the Fokker–Planck equation (FPE). The Fokker–Planck equation is a deterministic partial equation that describes how the probability density associated with a stochastic (randomly determined) process changes over time. In the context of deep learning, it provides a powerful method to analyze parameter distribution changes during training as they are influenced by both gradient-driven drift and stochastic diffusion.

This is simply because the Fokker–Planck equation can be directly derived from the SDE that defines this type of training. The SDE characterizes the evolution of a single parameter path, while the FPE describes how the probability density function (PDF) probed by those parameters evolves throughout parameter space. This dualism is crucial: SDEs function at the scale of single realizations while the FPE operates on that of ensembles. A study of the FPE provides information on global properties like convergence, stability and long-term behavior of deep learning models.

Conceptually, the Fokker–Planck equation is a combination of a drift term reflecting the deterministic attraction towards low loss regions and a diffusion term describing how probability spreads due to stochastic noise. The drift term does push the distribution at minima of the loss function, and the diffusion term prevents from collapsing in a point and enables spreading in pram space. This interplay of these two forces shapes the distribution and drives the temporal evolution.

This balance is critical for optimization, particularly in deep learning. If drift is effective, the distribution becomes very sharp for specific parameter values and leads to deterministic like behavior characteristic of full-batch gradient descent (Ferris et al., 2019). When diffusion prevails, the distribution disperses widely, leading to excessive randomness and slow convergence. The best training regime sits somewhere in the middle, where drift and diffusion are balanced to support both convergence and exploration. The Fokker–Planck equation creates a formalism for examining this balance, as well as how learning rate and batch size trade off with each other.

The concept of stationary distribution is one of the relevant concepts related to Fokker–Planck equation. A stationary distribution is a probability distribution that, as time passes, does not change; it describes the behaviour of the system over longer periods. In the context of SDE (stochastic differential equation) based training, stationary

distribution refers to the state where drift and diffusion effects balance each other out at equilibrium. This distribution is usually of a form that relates to exponential of negative loss function, making a direct link between optimization and statistical physics. Figuratively speaking, you are sampling from a distribution during training where lower-loss configurations are more likely, but not guaranteed.

Such a probabilistic interpretation also gives us some insight as to why stochastic optimization methods generalize well. Rather than converging to a single minimum, the model samples from a distribution of good solutions and preserves variability and uncertainty in parameter space. It results in more resilient models that generalize better to out-of-distribution data. So the Fokker–Planck framework naturally captures the relationship between stochasticity and generalization in a mathematically accurate way.

A further important point from the FPE perspective is that entropy governs the evolution of distributions. Diffusion implies a lower increase in the entropy of the parameter distribution leading to better exploration and diversity. In contrast, drift reduces entropy by resampling the high-probability areas with low loss. The balance of these competing effects dictates the entropy dynamics of the system. It is important to have an adequate level of entropy, as too much will stochastically prevent the model from pooling its trains in promising regions of loss landscape while having too little would lead to premature convergences.

We can also use the Fokker–Planck equation to analyze convergence rates and stability properties. One can compare the performance of optimization algorithms using the speed at which the probability distribution gets to its stationary state. Convergence behaviour is affected by noise structure, curvature of the loss landscape, and dimensionality of parameter space. This analysis offers insights and helpful recommendations for designing more efficient and stable training algorithms.

Component	Description	Role in Deep Learning Training
Drift Term	Positive (new words > old/ignored examples)	Guides parameters toward minima by following the average gradient.
Diffusion Term	Stochastic spreading due to noise	Promotes exploration of the loss landscape and prevents premature collapse.
Probability Density	Distribution of parameters over space	Represents the uncertainty and variability of the model weights.
Stationary Distribution	At equilibrium, drift and diffusion balance out	Defines the long-term behavior and final "resting state" of training.
Entropy Evolution	Evolution of uncertainty distribution	Balances the tension between broad exploration and final convergence.
Convergence Dynamics	SLOPE: Speed of approach for equilibrium in distribution	Determines training efficiency and how quickly the model stabilizes.

In particular, applying the Fokker–Planck equation exactly in practice is generally not possible because deep learning models are often of high dimensions. However, computational methods and simulations can give us an approximate idea about their distributional behavior. Such approximations also provide the freedom to study the evolution of parameter distributions as hyperparameters or architecture change in ways a trajectory-based analysis alone cannot illuminate.

The FPE framework also smoothly integrates with advanced topics such as Bayesian inference, variation methods, and optimal transport, where distributions — not just parameters — are what we want to study. The Fokker–Planck equation unifies the analyses of these domains, linking them together and casting a wide net over many areas in deep learning.

In summary, the Fokker–Planck equation gives an elegant and powerful framework to understand how the distribution of parameters evolves during training in deep learning. It can tell us a great deal about convergence, generalization and robustness by capturing the interplay between deterministic gradients and stochastic noise. By lifting the trajectory-based view offered by SDEs we are able to develop a full and nuanced understanding of training dynamics from a distributional perspective. The Fokker–Planck framework can therefore be expected to become an important foundation for the design of future more principled and efficient optimization techniques as deep learning progresses.

Stability of Training via SDEs

Training stability is the cornerstone of deep learning optimization, and often determines whether a model will be able converge to a useable solution or not during training. This necessity stems from the fact that now traditional analyses focused on deterministic systems where as modern training algorithms are inherently

stochastic and require richer frameworks. SDEs provide this architecture: they capture deterministic gradient flow — which may be viewed as an infinite dimensional approximation of a resonator — as well stochastic noise. SDE-based stability analysis provides a rigorous framework for understanding the temporal behavior of deep learning systems, including their resistance to perturbations and convergence properties to stable equilibrium.

A. Stability Concept for Stochastic Systems

Stability in deterministic systems is concerned with the question of whether small perturbations at time 0 (of the initial conditions) will result into small deviations on the trajectory over time. In the presence of stochasticity however, one must define stability in probabilistic terms. One studies not a single trajectory but the behavior of a distribution of such trajectories driven by uncertainty or noise. A stable system is one in which the probability distribution is bounded and becomes invariant as time approaches infinity.

In deep learning, this idea manifests itself in the form of how model parameters move during definitions and learning's. A stable training process will still ensure that even with noisy updates, parameters do not diverge but rather develop (evolve) to low loss regions. The SDE model, reveals how drift and diffusion act jointly to influence stability, a behavior we would like to formalize.

B. Lyapunov Stability in SDE Setting

Especially, one of the most useful characteristics that can be used to verify stability is Lyapunov functions, which describes energy (or potential) for systems. For deep learning, the loss function itself is often a natural Lyapunov function. A system is stable if the expected value of the loss becomes smaller and smaller as time continues.

Lyapunov analysis is further extended in SDEs to stochastic settings through an examination of the expected change in the Lyapunov function, under drift and diffusion. This expected change will be negative or bounded for a stable system. In this way, researchers are able to back out conditions under which training converges in the presence of noise.

Additionally, Lyapunov-based analysis can also explain why some optimization algorithms are cooler than others. Reliable training results are more likely to be produced by algorithms with an expected loss that continues to decrease in a controlled but stochastic fashion.

C. Noise-Induced Stabilization

Noise is disruptive in many physical processes; stochastic systems feature a variety of counterintuitive phenomena with noise-induced stabilization being an interesting example where randomness fosters stability rather than disrupts it. Over the gaps, deep learning can lose stability and slowly foray into more unstable regions of a network architecture landscape if stochastic noise are available. Noise pushes the system to are probably going towards flatter and more stable areas by repeatedly disturbing the parameters.

It is generally known that the latter effect plays a critical role in high-dimensional non-convex optimization problems, where deterministic methods can converge to unstable equilibrium. Noise creates variability that keeps the system moving, avoiding these regions, allowing it to find stronger solutions. This is interpreted as diffusion counterbalancing the effect of unstable drift from an SDE perspective.

Another way to understand noise-induced stabilization is that this is one of the reasons how stochastic gradient descent often tends to perform better than deterministic gradient descent in practice. The randomness of mini-batch sampling is a built-in mechanism through which stability and generalization are improved.

D. Convergence Behaviour in Stochastic Dynamics

Then, convergence in SDE-based training does not mean to achieve a single point but rather to arrive at a stable distribution! The noise ensures that parameters fluctuate even after equilibrium is reached. So, we have to define the convergence in terms of a stability by distribution, contrarily to primary point wise convergence.

The convergence speed is dictated by the strength of the drift, size of diffusion, and effectively also by local curvature in the loss landscape. Very strong drift speeds up the convergence but may hinder exploration and a very strong diffusion improves exploration, however slows down the convergence. Getting the balance between these forces right is vital for effective training.

SDE analysis: tools. Recent work has applied SDE techniques to provide uniform conditions under which training is guaranteed to converge. These Insights are instrumental for conceiving algorithms achieving both rapid convergence and robust performance.

E. Stability for deep networks in high-dimension

Your deep neural network lives in a very high dimensional parameter space, but there is little as people say stability for the higher dimensional spaces. Alternatively, under certain conditions different directions in parameter

space may have different stability properties for such parameters. Different directions converge quickly while others develop noise or curvature effects.

This complexity is captured in SDE modelling, namely with the use of anisotropic diffusion as noise varies among dimensions. That results in a subtle form of stability where the system is stable in some directions and unstable in others. This directional stability is at the heart of making sure global convergence occurs.

Furthermore, large models generally consist of interactions between multiple layers and components [2], making the stability analysis even more complicated. SDEs based approaches offer a systematic means to explore these interactions and pinpoint potential sources of instability.

F. Practical Implications for Training Algorithms

Insights of SDE-based stability analysis directly influence the design and tuning of training algorithms. One leading example of this is with learning rate selection and the stability it provides. Using too large learning rate can raise noise and cause divergence whereas small machinery generalization (SGD or mini-batch); typically, it results in slow convergence.

Another key aspect is batch size, as this determines the noise level of stochastic variation. Batch sizes create diffusion, which could give you the edge for exploration or stability depending on the degree of diffusion but bring another twist to; smaller tend to increase while larger batches decrease stability in general. By reducing noise, larger batch sizes can lead to more stable however less robust training.

You are also interpreting regularisation techniques like dropout and weight decay through the scope of SDEs as stabilization method. Augmenting noise or other mechanisms so that a trade-off is achieved between exploration and convergence.

Factor	Effect on Drift	Effect on Diffusion	Impact on Stability
Learning Rate	Scales the magnitude of gradient updates directly.	Amplifies stochastic noise (variance).	High → Unstable; Optimal → Stable convergence.
Batch Size	Minimal direct effect on the average gradient.	Controls noise magnitude (inverse relationship).	Smaller batch → Less stable, more exploratory.
Loss Landscape Curvature	Influences the direction and "pull" of the drift.	Modulates noise amplification based on local geometry.	High curvature → Potential instability or oscillation.
Regularization Methods	Alters the effective loss function (e.g., Weight Decay).	May introduce additional noise or dampen updates.	Enhances stability and improves generalization.
Model Complexity	Increases the dimensionality and complexity of the drift.	Increases the potential variability in diffusion.	More complex → Often more difficult to stabilize.

G. Summary and Insights

Stochastic differential equations offer a rich paradigm for studying the properties of deep learning training methods, with stability analysis in this framework being an important object of study. SDEs allow exact characterisation of convergence, robustness and sensitivity to perturbations by generalising classical stability concepts into stochastic settings. Key insights like Lyapunov stability, noise-induced stabilization and convergences in distribution reveal the constructive role of stochasticity for optimization.

Notably, stability of deep learning is driven not only by deterministic components, but also arises from the interplay between drift and diffusion. This interaction guides the path of training and decides if sensible solutions or not are reached. In summary, SDE-based analysis enables the design of training algorithms that are both stable and data-efficient for deep learning, which will lead to more reliable and scalable systems.

Stochastic Dynamics: A Tool for Generalization.

One of the most interesting and important properties of deep learning is generalization, which I define loosely as a model's ability to make accurate predictions on unseen data not contained in its training set. While modern neural networks are high capacity and over parameterized, they often generalize surprisingly well. This theoretical conflict cannot be fully captured by the traditional American statistical learning theory, especially in scenarios where non-convex optimization landscapes exist. This brings us to a better understanding when we develop the framework of training dynamics with stochastic differential equations (SDEs) which means randomness has an important and constructive role. Generalization is not only a function of model architecture or dataset size, but fundamentally depends on the stochastic nature of optimization.

You are working with deterministic optimization, like full-batch gradient descent, meaning that the training path is entirely determined by the gradient of your loss function. Although this method guarantees the solution lands in a local minimum, it frequently produces solutions which are overly sensitive to perturbation and over fit the training data. On the other hand, with stochastic optimization, we have training where we define a distribution of possible parameter trajectories (not just one deterministic path). Such a parameter uncertainty allows the model to search a wider region of the parameter space, which makes it more likely to find solutions that transfer well.

A well-studied mechanism by which stochastic dynamics affects generalization is the preference for flat minima over sharp minima.. Flat minima are areas in the loss landscape where the loss function doesn't change much under small perturbations of the parameters. This regions correspond to solutions resistant to input perturbation and model uncertainty, hence more likely to generalise on unseen examples. Within the SDE framework, it can be seen and interpreted at a high level in the same way as above: stochastic noise is an additional smoothing force that prevents converging to sharp minima. Sharp minima are minimally stable, meaning that perturbations will move the system away from these regions with certainty; noise would naturally push the system into wider and therefore less stable regions.

Another way to understand the relationship between noise and generalization is through implicit regularization. Implicit vs. Explicit Regularization – Unlike explicit regularization (like weight decay or dropout),implicit regularization arises inherently in the optimization process. The randomness created by mini-batch sampling regularizes the situation so that your model does not fit the training data as precisely. This noise penalizes a complex solution so the model searches for simpler and more generalizable representations. This behavior, in terms of SDE where the diffusion term spreads the parameter distribution and prevents it from collapsing into narrow regions of loss landscape, is captured here.

The distribution of parameters, as opposed to point estimates, is another crucial element related to the generalization in stochastic dynamics. The SDE framework posits that instead of converging to a single optimal parameter vector, training results in a distribution over parameters. This distribution captures the variability and uncertainty introduced by the stochastic updates. Models that can explicitly or implicitly sample from this distribution are able to represent many reasonable solutions, and therefore lead to less brittle predictions. In this interpretation, generalization is achieved by "integrating out" parameter uncertainty and is thus highly compatible with Bayesian approaches to machine learning.

Both the size and shape of noise is leveraged by NN based on generalization performance: Stochastic effects are induced via hyper-parameter selection like learning rate, batch size and gradient variance. Higher noise from smaller batch sizes encourages exploration and increases the chances of landing in flat minima. In comparison, higher batch sizes may help to reduce noise but they can push the minima sharper —this likely blades generalization. Also higher learning rates can exaggerate both drift and diffusion, which means more space in the loss landscape. These relationships emphasize the need to carefully tune hyperparameters to maintain an optimal balance between exploration and convergence.

The geometry of the loss landscape also interacts with stochastic dynamics to influence how generalization occurs. The high-dimensional landscape often has a number of interlinked flat regions whose interfaces are separated by thin walls. Stochastic noise allows the model to get over these walls, sampling from an entire network ofoύς solutions. That connectivity between flat regions implies that generalization is not related to a single optimal solution, but rather a family of solutions which maintain similar properties. This behavior can be naturally described using the SDE framework, which spells out that stochastic exploration is the mechanism by which these regions are found.

One other perspective of generalization comes from the idea of stability with respect to perturbations. A model that generalizes well defines outputs length till the reasonable bounds under slight variations in input data or coefficients. This stability is internally favoured by stochastic dynamics, which favours solutions that are less sensitive to perturbations. This is evident through the way parameters of the distribution locate in regions where the loss function is also smooth and stable, a phenomenon observed in SDE framework. Consequently, the model is able to maintain robustness against noise and different data variations in generalization performance.

This correspondence also extends to contemporary deep learning architectures such as convolutional neural networks and transformers. The other thing is that these models live in really high parameter spaces, and deterministic optimization would never have traversed the landscape properly. Because training is intrinsically stochastic, these models can traverse complex terrains and find generalizable solutions to problems, making them successful across broad application domains.

Recent research has found that the structure of stochastic noise is neither uniform nor inherent, but depends on both the data and model architecture. This structured noise can direct the optimization process in certain

directions, facilitating generalization that isotropic noise models cannot account for. The structure of this landscape is an active research area, and could lead to training algorithm designs that are more efficient and/or generalizable.

To wrap it all up, generalization in deep learning is literally coupled with stochastic dynamics problems, and that the SDE framework offers a solid basis to comprehend such bond. This enables us to represent the relevant phenomena responsible for generalization}, such as searching around the loss landscape, being biased to prefer flat minima, implicit regularization and robustness of deep learning methods against perturbations. These perspectives show that randomness is not necessarily an obstacle but instead plays a beneficial role in terms of the noise in deep learning. Instead, stochasticity becomes inherent to effective learning as models realize high performance in wired and uncertain designs.

Adaptive Optimization as Controlled SDE Systems

In most large-scale and complex learning tasks like deep neural networks, adaptive optimization algorithms such as Adam, RMSProp, and Ad grad have become indispensable tools in training. In comparison to standard stochastic gradient descent (SGD), which applies a same learning rate for all parameters, adaptive methods change grinding rates in real time based on historical gradient information. A Sufficient Condition An Input-driven Local Existence A Nonlocal Global Finiteness The Loss Dependent Solution of DODE based Learning (Engraft) 3 Controlled stochastic systems Recent proof that the goal-directed behavior can be interpreted in a particular view of controlled stochastic evolutionary dynamics, (Wang & Yoshi). Such viewpoint would offer a more comprehensive understanding of how adaptive optimizers shape training dynamics, stability and generalization properties.

A. Adaptive Optimisation as Modified Drift Dynamics

The drift term in the SDE formulation of deep learning captures the deterministic component of parameter updates given by gradients. Adaptive optimization algorithms change this drift by scaling gradients based on past data, such as moving averages of gradient size or variance. This led to preconditioned gradient flow, where effective learning rate of each parameter is set different. From an SDE perspective, this is merely changing the parameter space geometry. Compared to the steepest descent direction as tackled in Euclidean space, adaptive methods direct optimization into directions involving curvature and gradient variability. This allows for much more fast-moving around the loss landscape given that different parameters have various levels of sensitivity (particularly in high dimensional settings).

However, the drift modification also adds complexity. Even though adaptive methods can speed up convergence, they also tend to bias the optimization process towards certain regions of the parameter space which may lead to poor generalization. This trade-off is essential for making well-informed decisions in our training strategies.

B. Diffusion and Noise Control Over Time In control of time-varying diffusion noise

Besides, adaptive optimization algorithms act on the diffusion part of the SDE rather than just change it drift. This is a result of the fact that diffusion is determined entirely by gradients, so any mechanism which normalizes or rescales gradients will also determine in what form and how much noise enters the model. Adaptive methods correctly perform time-varying diffusion, where the degree of stochasticity changes during training. The normalization of gradient updates directly from these second-moment estimates in Adam is an example which minimizes and amplifies updates for directions with high and low variance respectively. This is what leads to a better compensated diffusion process where noise is more evenly divided across parameters. Such behavior can add stability and also avoid extreme change at training time.

Meanwhile, for exploration, it entails some control over diffusion. If we reduce noise too much, the model will be unable to effectively explore the loss landscape; if we add too much noise, the training becomes unstable. So adaptive methods, which focus on dynamically adjusting the diffusion with needed strongly stable convergence while making enough exploration.

C. Preconditioning and Geometry-Aware Optimization

Preconditioning is a hallmark of adaptive optimizers, which uses information about the scale and variance of gradients to transform them. Preconditioned in SDE framework can be thought as a change of metric of parameter space which implicitly changes the geometry for optimization.

Success relies on the geometry-aware optimization which allows the model to move more efficiently in low curvature directions while to avoid unstable updates in high curvature directions. This allows adaptive methods to converge faster than standard SGD in many problems where the loss landscape is more ill-conditioned.

Additionally since preconditioning is closely related to notions from information geometry, where you would like to optimize with respect to a space which mirrors the underlying structure of probability distributions.

Adaptive methods mimic this optimal behavior by using gradient statistics, leading to more efficient and stable convergence.

D. Stability and convergence under adaptive control

But controllability of the mechanism is important for both stability and convergence. These methods can dynamically vary learning rates and noise levels, which keeps training stable even in narrow scenarios. Adaptive optimizers serve this directly by preventing divergence in highly non-convex loss landscapes with high variance gradients, such as those found in deep networks.

So that, from an SDE viewpoint, describes drift-diffusion balancing such that it closes on a steady-state distribution. Adaptive control mechanisms allows for controlled fluctuations of the system, avoiding extremes but enabling exploration.

Nonetheless, the convergence properties of adaptive methods are often subtle. In some settings, they can even converge to diverging solutions with different generalization characteristics than SGD. This has motivated investigations of the theoretical properties of adaptive optimization and how they might influence learning dynamics.

E. Comparison with Standard SGD in the SDE Framework

Two measures of comparison based on adaptive optimization and standard SGD when viewed through the SDE framework. SGD depends on uniform gradient scaling and uses a relatively simple diffusion structure. This simplicity usually results in hardy generalization because the noise of SGD encourages exploration and implicit regularization.

However, adaptive methods have richer dynamics due to preconditioning and noise management. Although this can help improve the speed and stability of convergence, it may reduce some of the positive effects associated with introducing stochasticity. This contributes to the observed faster training and weaker generalization compared To SGD sometimes seen with adaptive optimizers.

The comparison of the above results emphasizes a trade-off between efficiency and generalizability, adding nuance to future works that choose an optimization method solely based on performance metrics.

F. Strategies based on Hybrid & Controlled SDE

Hybrid approaches that exploit the advantages of both SGD and adaptive methods have been investigated in recent works. These methods are essentially governed SDE systems with varying components of the optimization process changed dynamically to yield certain desired properties/comments. Such that training could begin using adaptive methods for faster convergence and later switch to SGD to attain better generalisation performance.

Another strategy is to control the diffusion term directly, adding noise explicitly during training in order to navigate the exploration-exploitation trade-off. These methods utilize the SDE framework to build both efficient and robust optimization algorithms.

These hybrid methods give evidence as to how optimization can be treated through such a controlled stochastic process. Through designed control mechanisms, we can achieve training dynamics for optimal performance.

G. Summary and Insights

Several modern deep learning optimization algorithms adaptively tune hyperparameters of the optimizers themselves, and they can be well understood in the framework of stochastic differential equations. With a combination of drift and diffusion terms these approaches define a controlled stochastic process over the optimization landscape. It enables the implementation of dynamic learning rates that enhances stability, and optimizes loss landscapes changing.

Meanwhile, the SDE perspective highlights a prominent trade-off for adaptive methods in model generalization and exploration. Although these methods can accelerate convergence and improve stability, one must be cautious that such approaches do not correspond to a loss of robustness.

This empowers a unified perspective in the analysis and design of higher-order training algorithms for adaptive optimization, where we treat adaptive optimization itself as a controlled SDE system. It improves its theoretical understanding and facilitates further research towards efficient and generalizable optimization algorithms in a wider range of deep learning environments.

Conclusion

By interpreting the dynamics of deep learning training from the perspective of stochastic differential equations (SDE), this work has provided a unified approach that blends together optimization, probability and

dynamical systems perspectives for the first time. In particular, hopping beyond the conventional discrete perspective of algorithms such as SGD, we show that continuous-time stochastic formulation would lead to clearer and more comprehensive understanding on convergence behavior, generalization, robustness and stability of neural networks. The SDE framework advances the theoretical groundwork and leads to practical insights regarding creating superior training strategies with greater robustness.

One of the main themes in this work is that deep learning optimization is fundamentally stochastic rather than deterministic. Indeed, the mini-batch sampling induces a randomness that is often seen as a negative point, and studies have shown that it actually plays a constructive role on training dynamics. Explicitly modelling such randomness via diffusion terms in SDEs provides a way to access the mechanisms through which models explore rich loss landscapes. In this dynamic process, the deterministic drift controlled by gradients is counterbalanced by stochastic diffusion driven by noise: it balances exploitation of known good solutions with exploration of new lines of descent. This balance is essential for the good efficacy of deep learning on high-dimensional, non-convex settings.

How this relates to our current deep neural networks from the SDE perspective, and generalization is one of the central authors' runs away. Instead of six close to the same optimizer point, distorted training processes lead to a distribution of parameter configurations. This distributional perspective shows that models prefer flat minima, which are robust against perturbations in the weight space, over sharp minima that where likely to over fit each particular dataset. The diffusion piece of SDEs inevitably pulls the system away from pointy areas and into exploring wider, smoother domains. This phenomenon explains the implicit regularization effect in SGD and related algorithms which suggest that noise plays a favourable role by simply learning.

Noise geometry and anisotropic diffusion make this understanding even more precise by establishing the (non-uniform, structured) nature of the stochasticity in deep learning. The stochastic gradient covariance introduces a noise that is dependent on the direction, and this interacts with the loss landscape curvature in behaviours of optimizing trajectory. The structured noise allows for navigation through difficult parameter spaces, enabling models to move away from unstable regions and converge on stable, generalizable solutions. The identification of anisotropic diffusion as an important condition in training dynamics is a major advance in the theoretical understanding of deep learning.

Another key contribution of this work is linking SDE-based modelling with Bayesian inference. The training can be understood as exhibiting the sampling of a posterior distribution over parameters through frameworks such as Lange in dynamics and stochastic gradient Lange in dynamics (SGLD). Such a probabilistic view is inherently able to represent uncertainty, facilitate model calibration, and also improve the robustness to distributional shifts. SDE, which combine optimization and probabilistic inference are a promising approach to provide a robust interpretability AI models.

Bringing the Fokker–Planck equation into play generalizes this analysis beyond single parameter trajectories to describe time evolution of entire probability distributions. The perspective this observation lends toward distributional convergence and stability emphasizes stationary distributions and the role played by entropy dynamics. Insights into the long-term behaviour of training processes — understanding how and under what conditions they settle down to an equilibrium point, can be obtained through analysis of the flow of probability mass through parameter space [5]. This is a complementary approach to the trajectory-based view of SDEs, and together with it provides an overarching framework for studying the dynamics of deep learning.

The importance of noise for robust training has also been emphasized by stability analysis in SDEs. Examples such as Lyapunov stability and noise-induced stabilization show the opposite: such stochastic structures are stabilizing rather than destabilizing. The presence of noise plays the role of a stabilizing factor in deep learning optimization process as it causes to avoid becoming stuck in unstable regions while allowing the region's subsequent convergence toward stable equilibrium. It is a very different result, as it questions the usual belief that randomness is the best thing and highlights the need to control stochastic effects more carefully while training algorithms. We showed that adaptive optimization methods in the SDE framework and modern algorithms such as Adam or RMSProp can be seen as periodically controlled stochastic systems. These methods incorporate dynamic adjustment on either the drift or diffusion components of the stochastic process to add more flexibility, thereby enhancing convergence speed and stability during optimization. But there are also trade-offs between stability and generalization, indicating how a two-pronged approach (balancing adaptive with non-adaptive methods) might be prudent.

Although SDE-based modelling has enabled many important advances, there are still a number of challenges to tackle. This makes sense because to accurately capture the full picture of deep learning dynamics in high-dimensional parameter spaces is inherently hard, and many theoretical findings treat simplified assumptions. Additionally, continuous-time models cannot be practically used due to their computational constraints. Yet, ample

work on numerical methods, approximation techniques and scalable algorithms still addresses these obstacles to further scale up the applicability of SDE-based methods.

The future potential of this work lies in the combination these stochastic differential equation modelling techniques with new areas such as diffusion models, neural ordinary differential equations, and physics-informed machine learning. Two main implications of these developments are that the future of deep learning is increasingly anchoring on continuous-time and probabilistic frameworks, allowing for more principled and flexible learning from data. With the insights learned from SDE analysis, scientists may construct algorithms that are efficient in nature and scale according to real world situations.

We report the present study as setting up stochastic differential equations as a central dogma of deep learning training dynamics. SDEs provide a unified description that relates the deterministic nature of optimization and the stochastic aspect of exploring by integrating two major facets in machine learning (i.e., generalization/robustness and uncertainty/stability). The framework offers both theoretical insight and concrete guidance for translating the models into practical improvements. As deep learning models matured, so should the understanding of the limitations, and the guidance that SDE-based modelling will provide in finding ways to construct the new generation of intelligent systems designed for operation in an increasingly complex and uncertain environment.

References

[1] Y. Lacuna, Y. Bagnio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[2] I. Good fellow, Y. Bagnio, and A. Carville, *Deep Learning*. MIT Press, 2016.
[3] S. Ruder, "An overview of gradient descent optimization algorithms," *arrive preprint arXiv:1609.04747*, 2016.
[4] L. Bottom, F. E. Curtis, and J. Necedah, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.
[5] H. Risen, *The Fokker–Planck Equation: Methods of Solution and Applications*. Springer, 1996.
[6] B. Kendal, *Stochastic Differential Equations: An Introduction with Applications*. Springer, 2003.
[7] D. J. Hingham, "An algorithmic introduction to numerical simulation of stochastic differential equations," *SIAM Review*, vol. 43, no. 3, pp. 525–546, 2001.
[8] M. Welling and Y. W. The, "Bayesian learning via stochastic gradient Lange in dynamics," in *Proc. ICML*, 2011.
[9] M. Mandy, M. D. Hoffman, and D. M. Bali, "Stochastic gradient descent as approximate Bayesian inference," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 4873–4907, 2017.
[10] X. Li, T. Chen, H. Girolami, and J. A. Turner, "Stochastic gradient Hamiltonian Monte Carlo," in *Proc. ICML*, 2014.
[11] S. Goal et al., "Accurate, large minibatch SGD: Training Image Net in 1 hour," *arXiv:1706.02677*, 2017.
[12] S. Chaudhari and S. Sotto, "Stochastic gradient descent performs variation inference," *arXiv:1710.11029*, 2017.
[13] J. Sirignano and K. Iliopoulos, "Mean field analysis of neural networks," *SIAM J. Appl. Math.*, vol. 80, no. 2, pp. 725–752, 2020.
[14] J. Weaning E, "A proposal on machine learning via dynamical systems," *Communications in Mathematics and Statistics*, vol. 5, pp. 1–11, 2017.
[15] R. Ten and M. Raginsky, "Theoretical guarantees for sampling and inference in generative models with latent diffusions," *COLT*, 2019.
[16] Y. Li, T. Ma, and H. Zhang, "Learning over parameterized neural networks via stochastic gradient descent," *arXiv:1807.00263*, 2018.
[17] N. S. Cesar et al., "On large-batch training for deep learning: Generalization gap and sharp minima," *ICLR*, 2017.
[18] S. Hoch Reiter and J. Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.
[19] P. Chaudhari et al., "Entropy-SGD: Biasing gradient descent into wide valleys," *ICLR*, 2017.
[20] J. Zhang et al., "Understanding deep learning requires rethinking generalization," *ICLR*, 2017.
[21] D. P. Kingman and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.
[22] T. Tie leman and G. Hinton, "RMSProp: Divide the gradient by a running average," *Neural Networks for Machine Learning*, 2012.
[23] A. Krizhevsky, I. Sutskever, and G. Hinton, "Image Net classification with deep convolutional neural networks," *NIPS*, 2012.
[24] A. Aswan et al., "Attention is all you need," *NIPS*, 2017.
[25] R. Neal, *Bayesian Learning for Neural Networks*. Springer, 1996.
[26] C. Gardiner, *Handbook of Stochastic Methods*. Springer, 2004.
[27] T. Chen et al., "Neural ordinary differential equations," *Neutrals*, 2018.
[28] J. Song and S. Sermon, "Generative modelling by estimating gradients of the data distribution," *Neutrals*, 2019.
[29] Y. Song et al., "Score-based generative modelling through stochastic differential equations," *ICLR*, 2021.
[30] M. Raginsky, A. Sakhalin, and M. Telgarsky, "Non-convex learning via stochastic gradient Lange in dynamics," *COLT*, 2017.
[31] L. Sagan, L. Bottom, and Y. Lacuna, "Eigenvalues of the Hessian in deep learning," *arXiv:1611.07476*, 2016.
[32] A. Yastrzemski et al., "Three factors influencing minima in SGD," *ICLR Workshop*, 2018.
[33] K. Kawaguchi, "Deep learning without poor local minima," *Neutrals*, 2016.
[34] S. Malady et al., "On the diffusion approximation of SGD," *arXiv:2006.05987*, 2020.
[35] A. S. Sznitman, "Topics in propagation of chaos," *Springer Lecture Notes in Mathematics*, 1991.