SYMMETRY, CONSERVATION LAWS, AND LEARNING DYNAMICS IN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting the dynamics of neural network parameters during training is one of the key challenges in building a theoretical foundation for deep learning. A central obstacle is that the motion of a network in high-dimensional parameter space undergoes discrete finite steps along complex stochastic gradients derived from real-world datasets. We circumvent this obstacle through a unifying theoretical framework based on intrinsic symmetries embedded in a network's architecture that are present for *any* dataset. We show that any such symmetry imposes stringent geometric constraints on gradients and Hessians, leading to an associated conservation law in the continuous-time limit of stochastic gradient descent (SGD), akin to Noether's theorem in physics. We further show that finite learning rates used in practice can actually break these symmetry induced conservation laws. We apply tools from finite difference methods to derive *modified gradient flow*, a differential equation that better approximates the numerical trajectory taken by SGD at finite learning rates. We combine modified gradient flow with our framework of symmetries to derive exact integral expressions for the dynamics of certain parameter combinations. We empirically validate our analytic predictions for learning dynamics on VGG-16 trained on Tiny ImageNet. Overall, by exploiting symmetry, our work demonstrates that we can analytically describe the learning dynamics of various parameter combinations at finite learning rates and batch sizes for state of the art architectures trained on *any* dataset.

1 INTRODUCTION

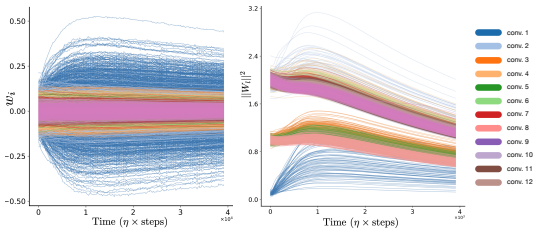
Just like the fundamental laws of classical and quantum mechanics taught us how to control and optimize the physical world for engineering purposes, a better understanding of the laws governing neural network learning dynamics can have a profound impact on the optimization of artificial neural networks. This raises a foundational question: *what, if anything, can we quantitatively predict about the learning dynamics of large-scale, non-linear neural network models driven by real-world datasets and optimized via stochastic gradient descent with a finite batch size, learning rate, and with or without momentum?* In order to make headway on this extremely difficult question, existing works have made major simplifying assumptions on the network, such as restricting to identity activation functions Saxe et al. (2013), infinite width layers Jacot et al. (2018), or single hidden layers Saad & Solla (1995). Many of these works have also ignored the complexity introduced by stochasticity and discretization by only focusing on the learning dynamics under gradient flow. In the present work, we make the first step in an orthogonal direction. Rather than introducing unrealistic assumptions on the model or learning dynamics, we uncover restricted, but meaningful, combinations of parameters with simplified dynamics that can be solved exactly without introducing a single assumption (see Fig. 1). To find the parameter combinations, we use the lens of *symmetry* to show that if the training loss doesn't change under some transformation of the parameters, then the gradient and Hessian for those parameters have associated geometric constraints. We systematically apply this approach to modern neural networks to derive exact integral expressions and verify our predictions empirically on large scale models and datasets. We believe our work is the first step towards a foundational understanding of neural network learning dynamics that is not based in simplifying assumptions, but rather the simplifying symmetries embedded in a network's architecture. Our main contributions are:

1. We leverage continuous differentiable symmetries in the loss to unify and generalize geometric constraints on neural network gradients and Hessians (section 3).

2. We prove that each of these differentiable symmetries has an associated conservation law under the learning dynamics of gradient flow (section 4).
3. We construct a more realistic continuous model for stochastic gradient descent by modeling weight decay, momentum, stochastic batches, and finite learning rates (section 5).
4. We show that under this more realistic model the conservation laws of gradient flow are broken, yielding simple ODEs governing the dynamics for the previously conserved parameter combinations (section 6).
5. We solve these ODEs to derive exact learning dynamics for the parameter combinations, which we validate empirically on VGG-16 trained on Tiny ImageNet with and without batch normalization (section 6).

2 Related Work

The goal of this work is to construct a theoretical framework to better understand the learning dynamics of state-of-the-art neural networks trained on real-world datasets. Existing works have made progress towards this goal through major simplifying assumptions on the architecture and learning rule. Saxe et al. (2013) considered linear neural networks with specific orthogonal initializations, deriving exact solutions for the learning dynamics under gradient flow. The theoretical tractability of linear networks has further enabled analyses on the properties of loss landscapes Kawaguchi (2016), convergence Arora et al. (2018a); Du & Hu (2019), and implicit acceleration by overparameterization Arora et al. (2018b). Saad & Solla (1995) studied a single hidden layer architecture with non-linearities in a student-teacher setup, deriving a set of complex ODEs describing the learning dynamics. Such shallow neural networks have also catalyzed recent major advances in understanding convergence properties of neural networks Du et al. (2018b); Mei et al. (2018). Jacot et al. (2018) considered infinitely wide neural networks with non-linearities, demonstrating that the network's prediction becomes linear in its parameters. This setting allows for an insightful mathematical formulation of the network's learning dynamics as a form of kernel regression where the kernel is defined by the initialization. Arora et al. (2019) extended these results to convolutional networks and Lee et al. (2019) demonstrated how this understanding also allows for predictions of parameter dynamics.



(a) Parameter Dynamics (b) Neuron Dynamics

Figure 1: **Neuron level dynamics are simpler than parameter dynamics.** We plot the per-parameter dynamics (left) and per-channel squared Euclidean norm dynamics (right) for the convolutional layers of a VGG-16 model (with batch normalization) trained on Tiny ImageNet with SGD with learning rate $\eta = 0.1$, weight decay $\lambda = 10^{-4}$, and batch size $S = 256$. While the parameter dynamics are noisy and chaotic, the neuron dynamics are smooth and patterned.

In the present work, we make the first step in an orthogonal direction. Instead of introducing unrealistic assumptions, we discover restricted combinations of parameters for which we can find exact solutions, as shown in Fig. 1. We make this fundamental contribution by constructing a framework harnessing the geometry of the loss shaped by symmetry and realistic continuous equations of learning.

Geometry of the loss. A wide range of literature has discussed constraints on gradients originating from specific architectural building blocks of networks. For the first part of our work, we simplify, unify, and generalize the literature through the lens of symmetry.

The earliest works understanding the importance of invariances in neural networks come from the loss landscape literature Baldi & Hornik (1989) and the characterization of critical points in the presence of explicit regularization Kunin et al. (2019). More recent works have studied implicit regularization originating from linear Arora et al. (2018b) and homogeneous Du et al. (2018a) activations, finding that gradient geometry plays an important role in constraining the learning dynamics. A different line of research studying the generalization capacity of networks have noticed similar gradient structures Liang et al. (2019). Beyond theoretical studies, geometric properties of the gradient and Hessian have

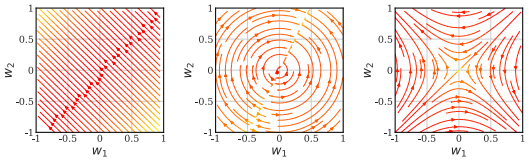
been applied to optimize Neyshabur et al. (2015), interpret Bach et al. (2015), and prune Tanaka et al. (2020) neural networks.

Gradient properties introduced by batch Ioffe & Szegedy (2015), weight Salimans & Kingma (2016) and layer Ba et al. (2016) normalization have been intensely studied. Van Laarhoven (2017); Zhang et al. (2018) showed that normalization layers are scale invariant, but have an implicit role in controlling the effective learning rate. Cho & Lee (2017); Hoffer et al. (2018); Chiley et al. (2019); Li & Arora (2019); Wan et al. (2020) have leveraged the scale invariance of batch normalization to understand geometric properties of the learning dynamics. Most recently, Li et al. (2020) studied the role of gradient noise in reconciling the empirical dynamics of batch normalization with the theoretical predictions given by continuous models of gradient descent.

Equations of learning. To make experimentally testable predictions on learning dynamics, we introduce a continuous model for stochastic gradient descent (SGD). There exists a large body of works studying this subject using stochastic differential equations (SDEs) in the continuous-time limit Mandt et al. (2015; 2017); Li et al. (2017); Smith & Le (2017); Chaudhari & Soatto (2018); Jastrzębski et al. (2017); Zhu et al. (2018); An et al. (2018). Each of these works involves making specific assumptions on the loss or the noise in order to derive stationary distributions. More careful treatment of stochasticity led to fluctuation dissipation relationships at steady state without such assumptions Yaida (2018). In our analysis, we apply a more recent approach Li et al. (2017); Barrett & Dherin (2020), inspired by finite difference methods, that augments SDE model with higher-order terms to account for the effect of a finite step size and curvature in the learning trajectory.

3 Symmetries in the Loss shape Gradient and Hessian Geometries

While we initialize neural networks randomly, their gradients and Hessians at all points in training, no matter the loss or dataset, obey certain geometric constraints. Some of these constraints have been noticed previously as a from of implicit regularization, while others have been leveraged algorithmically in applications from network pruning to interpretability. Remarkably, all these geometric constraints can be understood as consequences of numerous differentiable symmetries in the loss introduced by neural network architectures. A set of parameters observes a differentiable symmetry in the loss if the loss doesn't change under a certain differentiable transformation of these parameters. This invariance introduces associated geometric constraints on the gradient and Hessian.



(a) Translation (b) Scale (c) Rescale

Figure 2: **Visualizing symmetry.** We visualize the vector fields associated with simple network components that have translation, scale, and rescale symmetry. In (a) we consider the vector field associated with a neuron $\sigma\left(\begin{bmatrix} w_1 & w_2 \end{bmatrix}^\intercal x\right)$ where σ is the softmax function. In (b) we consider the vector field associated with a neuron $\text{BN}\left(\begin{bmatrix} w_1 & w_2 \end{bmatrix}\begin{bmatrix} x_1 & x_2 \end{bmatrix}^\intercal\right)$ where BN is the batch normalization function. In (c) we consider the vector field associated with a linear path $w_2 w_1 x$.

Consider a function $f(\theta)$ where $\theta \in \mathbb{R}^m$. This function possesses a differentiable symmetry if it is invariant under the differentiable action ψ of a group G on the parameter vector θ, i.e., if $\theta \mapsto \psi(\theta, \alpha)$ where $\alpha \in G$, then $F(\theta, \alpha) = f(\psi(\theta, \alpha)) = f(\theta)$ for all (θ, α). The existence of a symmetry enforces a geometric structure on the gradient, ∇F. Evaluating the gradient at the identity element of G, so that $\psi(\theta, \alpha) = \theta$, yields the result,

$$\langle \nabla f, \partial_\alpha \psi \rangle = 0, \tag{1}$$

which implies that the gradient ∇f is perpendicular to the vector field $\partial_\alpha \psi$ that generates the symmetry, for all θ. The symmetry also enforces a geometric structure on the Hessian, $\mathbf{H}F$. Evaluating the Hessian at the identity element of G yields the result,

$$\mathbf{H}f \partial_\theta \psi \partial_\alpha \psi + \partial_\theta \partial_\alpha \psi \nabla f = 0, \tag{2}$$

which constrains the Hessian $\mathbf{H}f$. See appendix A for the derivation of these properties and other geometric consequences of symmetry.

We will now consider the specific setting of a neural network parameterized by $\theta \in \mathbb{R}^m$, the training loss $\mathcal{L}(\theta)$, and three families of symmetries (translation, scale, and rescale) that commonly appear in modern network architectures.

Translation symmetry. Translation symmetry is defined by the group $\mathbb{R}$ and action $\psi_\alpha(\theta) = \theta + \alpha \mathbb{1}_\mathcal{A}$ where $\mathbb{1}_\mathcal{A}$ is the indicator vector for some subset $\mathcal{A}$ of the parameters $\{\theta_1, \ldots, \theta_m\}$. The loss function $\mathcal{L}$ thus possesses translation symmetry if $\mathcal{L}(\theta) = \mathcal{L}(\theta + \alpha \mathbb{1}_\mathcal{A})$ for all $\alpha \in \mathbb{R}$. Such a symmetry in turn implies the loss gradient $\partial_\theta \mathcal{L} = g$ is orthogonal to the indicator vector $\partial_\alpha \psi_\alpha = \mathbb{1}_\mathcal{A}$,

$$\langle g, \mathbb{1}_\mathcal{A} \rangle = 0, \tag{3}$$

and that the Hessian matrix $H = \partial_\theta^2 \mathcal{L}$ has the indicator vector in its kernel,

$$H \mathbb{1}_\mathcal{A} = 0. \tag{4}$$

Softmax function. Any network using the softmax function gives rise to translation symmetry for the parameters immediately preceding the function. Let $z = Wx + b$ be the input to the softmax function such that $\sigma(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$. Notice that shifting any column of the weight matrix w_i or the bias vector b by a real constant has no effect on the output of the softmax as the shift factors from both the numerator and denominator canceling its effect. Thus, the loss function is invariant w.r.t. this translation, yielding the gradient constraints $\langle \frac{\partial \mathcal{L}}{\partial w_i}, \mathbb{1} \rangle = \langle \frac{\partial \mathcal{L}}{\partial b}, \mathbb{1} \rangle = 0$, visualized in Fig. 2 for the toy model where $w_i \in \mathbb{R}^2$.

Scale symmetry. Scale symmetry is defined by the group $\mathrm{GL}_1^+(\mathbb{R})$ and action $\theta \mapsto \alpha_\mathcal{A} \odot \theta$ where $\alpha_\mathcal{A} = \alpha \mathbb{1}_\mathcal{A} + \mathbb{1}_{\mathcal{A}^c}$. The loss function possesses scale symmetry if $\mathcal{L}(\theta) = \mathcal{L}(\alpha_\mathcal{A} \odot \theta)$ for all $\alpha \in \mathrm{GL}_1^+(\mathbb{R})$. This symmetry immediately implies the loss gradient is everywhere perpendicular to the parameter vector itself $\partial_\alpha \psi_\alpha = \theta \odot \mathbb{1}_\mathcal{A} = \theta_\mathcal{A}$,

$$\langle g, \theta_\mathcal{A} \rangle = 0, \tag{5}$$

and relates to the Hessian matrix, where $\partial_\theta \mathcal{L} \partial_\alpha \partial_\theta \psi_\alpha = g \ \mathrm{diag}(\mathbb{1}_\mathcal{A}) = g_\mathcal{A}$, as

$$H \theta_\mathcal{A} + g_\mathcal{A} = 0. \tag{6}$$

Batch normalization. Batch normalization leads to scale invariance during training. Let $z = w^\intercal x + b$ be the input to a neuron with batch normalization such that $\mathrm{BN}(z) = \frac{z - \mathrm{E}[z]}{\sqrt{\mathrm{Var}(z)}}$ where $\mathrm{E}[z]$ is the sample mean and $\mathrm{Var}(z)$ is the sample variance given a batch of data. Notice that scaling w and b by a non-zero real constant has no effect on the output of the batch normalization as it factors from z, $\mathrm{E}[z]$, and $\mathrm{Var}(z)$ canceling its effect. Thus, these parameters observe scale symmetry in the loss and their gradients satisfy $\langle \frac{\partial \mathcal{L}}{\partial w}, w \rangle + \langle \frac{\partial \mathcal{L}}{\partial b}, b \rangle = 0$, as has been previously noted by Ioffe & Szegedy (2015); Van Laarhoven (2017), and visualized in Fig. 2 for the toy model where $w, b \in \mathbb{R}$.

Rescale symmetry. Rescale symmetry is defined by the group $\mathrm{GL}_1^+(\mathbb{R})$ and action $\theta \mapsto \alpha_{\mathcal{A}_1} \odot \alpha_{\mathcal{A}_2}^{-1} \odot \theta$ where $\mathcal{A}_1$ and $\mathcal{A}_2$ are two disjoint sets of parameters. The loss function possesses rescale symmetry if $\mathcal{L}(\theta) = \mathcal{L}(\alpha_{\mathcal{A}_1} \odot \alpha_{\mathcal{A}_2}^{-1} \odot \theta)$ for all $\alpha \in \mathrm{GL}_1^+(\mathbb{R})$. This symmetry immediately implies the loss gradient is everywhere perpendicular to the sign inverted parameter vector $\partial_\alpha \psi = \theta_{\mathcal{A}_1} - \theta_{\mathcal{A}_2} = \theta \odot (\mathbb{1}_{\mathcal{A}_1} - \mathbb{1}_{\mathcal{A}_2})$,

$$\langle g, \theta_{\mathcal{A}_1} - \theta_{\mathcal{A}_2} \rangle = 0 \tag{7}$$

and relates to the Hessian matrix, where $\partial_\theta \mathcal{L} \partial_\alpha \partial_\theta \psi = g \mathrm{diag}(\mathbb{1}_{\mathcal{A}_1} - \mathbb{1}_{\mathcal{A}_2}) = g_{\mathcal{A}_1} - g_{\mathcal{A}_2}$, as

$$H(\theta_{\mathcal{A}_1} - \theta_{\mathcal{A}_2}) + g_{\mathcal{A}_1} - g_{\mathcal{A}_2} = 0. \tag{8}$$

Homogeneous activation. For networks with continuous, homogeneous activation functions $\phi(z) = \phi'(z) z$ (e.g. ReLU, Leaky ReLU, linear), this symmetry emerges at every hidden neuron by considering all incoming and outgoing parameters to the neuron. For example, consider a hidden neuron with ReLU activation $\phi(z) = \max\{0, z\}$, such that $w_2 \phi(w_1^\intercal x + b)$ is the computational path through this neuron. Scaling w_1 and b by a real constant and w_2 by its inverse has no effect on the computational path as the constants can be passed through the ReLU activation canceling their effects. Thus, these parameters observe rescale symmetry and their gradients satisfy $\left\langle \frac{\partial \mathcal{L}}{\partial w_1}, w_1 \right\rangle + \langle \frac{\partial \mathcal{L}}{\partial b}, b \rangle - \left\langle \frac{\partial \mathcal{L}}{\partial w_2}, w_2 \right\rangle = 0$, as has been previously noted by Du et al. (2018a); Liang

et al. (2019); Tanaka et al. (2020), and visualized in Fig. 2 for the toy model where $w_1, w_2 \in \mathbb{R}$ and $b = 0$.

General rescale symmetry. A more general rescale symmetries can be defined by the group $\mathrm{GL}_1^+(\mathbb{R})$ and action $\theta \mapsto \alpha_{\mathcal{A}_1}^{k_1} \odot \alpha_{\mathcal{A}_2}^{-k_2} \odot \theta$, which occur in networks with quadratic activation functions. A stronger form of rescale symmetry occurs for linear networks under the action of the group $GL_k^+(\mathbb{R})$ of $k \times k$ invertible matrices, as noticed previously by Arora et al. (2018a); Du et al. (2018a).

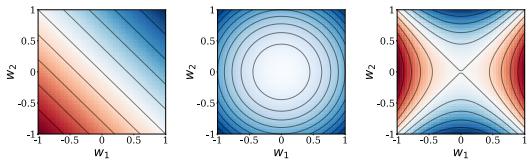
4 Symmetry leads to Conservation Laws under Gradient Flow

We now explore how geometric constraints on gradients and Hessians arising as a consequence of symmetry, impact network learning dynamics. In this work we focus on stochastic gradient descent (SGD), the workhorse of modern deep learning optimization. We will consider a model parameterized by θ, a training dataset $\{x_1, \ldots, x_N\}$ of size N, and a training loss $\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(\theta, x_i)$ with corresponding gradient $g(\theta) = \frac{\partial \mathcal{L}}{\partial \theta}$.

The gradient descent update with learning rate η is $\theta^{(n+1)} = \theta^{(n)} - \eta g(\theta^{(n)})$, which is a forward Euler discretization[1] with step size η of the ordinary differential equation (ODE)

$$\frac{d\theta}{dt} = -g(\theta). \tag{9}$$

In the limit as $\eta \to 0$, gradient descent exactly matches the dynamics of this ODE, which is commonly referred to as gradient flow Kushner & Yin (2003). Equipped with a continuous model for the learning dynamics, we now ask how do the dynamics interact with the geometric properties introduced by symmetries?



(a) Translation (b) Scale (c) Rescale

Figure 3: **Visualizing conservation.** Associated with each symmetry is a conserved quantity constraining the gradient flow dynamics to a surface. For translation symmetry (a) the flow is constrained to a hyperplane where the intercept is conserved. For scale symmetry (b) the flow is constrained to a sphere where the radius is conserved. For rescale symmetry (c) the flow is constrained to a hyperbola where the axes are conserved. The color represents the value of the conserved quantity, where blue is positive and red is negative, and the black lines are level sets.

Symmetry leads to conservation. Strikingly similar to Noether's theorem, which describes a fundamental relationship between symmetry and conservation for physical systems governed by Lagrangian dynamics, *every symmetry of a network architecture has a corresponding conserved quantity through training under gradient flow*. Consider some subset of the parameters $\mathcal{A}$ that respects either a translation, scale, or rescale symmetry. As shown in section 3, the gradient of the loss $g(\theta)$ is always perpendicular to the vector field that generates the symmetry $\partial_\alpha \psi$. Projecting the gradient flow learning dynamics in equation (27) onto the generator vector field yields a differential equation $\langle \frac{d\theta}{dt}, \partial_\alpha \psi \rangle = 0$. Integrating this equation through time results in the conservation laws,

$$\textbf{Translation:} \quad \langle \theta_{\mathcal{A}}(t), \mathbb{1} \rangle = \langle \theta_{\mathcal{A}}(0), \mathbb{1} \rangle \tag{10}$$

$$\textbf{Scale:} \quad |\theta_{\mathcal{A}}(t)|^2 = |\theta_{\mathcal{A}}(0)|^2 \tag{11}$$

$$\textbf{Rescale:} \quad |\theta_{\mathcal{A}_1}(t)|^2 - |\theta_{\mathcal{A}_2}(t)|^2 = |\theta_{\mathcal{A}_1}(0)|^2 - |\theta_{\mathcal{A}_2}(0)|^2 \tag{12}$$

Each of these equations define a conserved constant of learning through training. For parameters with translation symmetry, their sum is conserved, effectively constraining their dynamics to a hyperplane. For parameters with scale symmetry, their euclidean norm is conserved, effectively constraining their dynamics to a sphere. For parameters with rescale symmetry, their difference in squared euclidean norm is conserved, effectively constraining their dynamics to a hyperbola. In Fig. 3 we visualize the level sets of these conserved quantities for the toy models discussed in Fig. 2.

[1] See appendix B for a detailed derivation.

5 A REALISTIC CONTINUOUS MODEL FOR STOCHASTIC GRADIENT DESCENT

In section 4 we combined the geometric constraints introduced by symmetries with gradient flow to derive conservation laws for simple combinations of parameters during training. However, empirically we know these laws are broken, as demonstrated in Fig. 1. What causes this discrepancy? Gradient flow is too simple of a continuous model for realistic SGD training. It fails to incorporate the effect of weight decay introduced by explicit regularization, momentum introduced by commonly used hyperparameters, stochasticity introduced by random batches, and discretization introduced by a finite learning rate. Here, we construct a more realistic continuous model for stochastic gradient descent.

Modeling weight decay. Explicit regularization through the addition of an L_2 penalty on the parameters, with regularization constant λ, is very common practice when training modern deep learning models. This is generally implemented not by modifying the training loss, but rather directly modifying the optimizers update equation. For stochastic gradient descent, the result leads to the updated continuous model

$$\frac{d\theta}{dt} = -g(\theta) - \lambda\theta. \tag{13}$$

Modeling momentum. Momentum is a common extension to SGD that uses an exponentially moving average of gradients to update parameters rather than a single gradient evaluation Rumelhart et al. (1986). The method introduces two additional hyperparameters, a damping coefficient α and a momentum coefficient β, and applies the two step update equation, $\theta^{(n+1)} = \theta^{(n)} - \eta v^{(n+1)}$ where $v^{(n+1)} = \beta v^{(n)} + (1-\alpha) g(\theta^{(n)})$. When $\alpha = \beta = 0$, we regain classic gradient descent. In general, α effectively reduces the learning rate and β controls how past gradients are used in future updates resulting in a form of "inertia" accelerating and smoothing the descent trajectory. Rearranging the two-step update equation, we find that gradient descent with momentum is a first-order discretization with step size $\eta(1-\alpha)$ of the ODE,

$$(1-\beta)\frac{d\theta}{dt} = -g(\theta). \tag{14}$$

A more detailed derivation for this ODE can be found in appendix B.

Modeling stochasticity. Stochastic gradients $\hat{g}_{\mathcal{B}}(\theta)$ arise when we consider a batch $\mathcal{B}$ of size S drawn uniformly from the indices $\{1, \ldots, N\}$ forming the unbiased gradient estimate $\hat{g}_{\mathcal{B}}(\theta) = \frac{1}{S}\sum_{i\in\mathcal{B}} \nabla \ell(\theta, x_i)$. When the batch size is much smaller than the size of the dataset, $S \ll N$, then we can model the batch gradient as an average of S i.i.d. samples from a noisy version of the true gradient $g(\theta)$. Using the central limit theorem, we assume $\hat{g}_{\mathcal{B}}(\theta) - g(\theta)$ is a Gaussian random variable with mean $\mu = 0$ and covariance matrix $\Sigma = \frac{1}{S} G(\theta) G(\theta)^\intercal$. Under this assumption, the stochastic gradient update can be written as $\theta^{(n+1)} = \theta^{(n)} - \eta g(\theta^{(n)}) + \frac{\eta}{\sqrt{S}} G(\theta)\xi$, where ξ is a standard normal random variable. This update is an Euler-Maruyama discretization with step size η of the stochastic differential equation

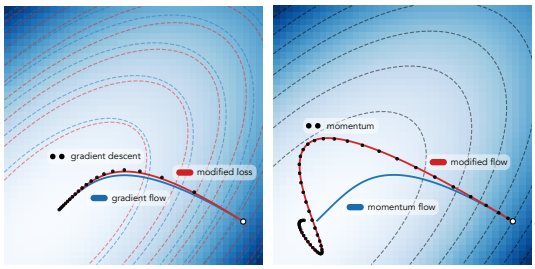
$$d\theta = -g(\theta)dt + \sqrt{\frac{\eta}{S}} G(\theta) dW_t, \tag{15}$$

where W_t is a standard Wiener process. Many stochastic differential equation approximations for SGD similar to equation (15) have been derived in previous works. In order to simplify the analysis, many of these works have then made additional assumptions on the covariance matrix $\Sigma = G(\theta)G(\theta)^\intercal$, such as $\Sigma = H$ where H is the Hessian matrix Jastrzębski et al. (2017), $\Sigma = C$ where C is some constant matrix Mandt et al. (2015), and $\Sigma = I$ where I is the identity matrix Chaudhari & Soatto (2018). However, without *any* additional assumptions, the differential symmetries intrinsic to neural network architectures add fundamental constraints on Σ.

As we showed in section 3, the gradient of the loss, regardless of the batch, is orthogonal to the generator vector field $\partial_\alpha \psi$ associated with a symmetry. This implies the stochastic noise must also observe the same property, $\langle -\frac{1}{\sqrt{S}} G(\theta)\xi, \partial_\alpha \psi \rangle = 0$. In order for this relationship to hold for arbitrary noise[2] ξ, then $G(\theta)^\intercal \partial_\alpha \psi = 0$. In other words, the differential symmetry inherent in neural network architectures projects the noise introduced by stochastic gradients to low rank subspaces.

[2] We do not need to assume the noise is Gaussian in order for this property to be true. However, we adopt this commonly accepted assumption to contextualize our work within the literature modeling SGD.

Modeling discretization. The effect of discretization when modeling continuous dynamics is a well studied problem in the numerical analysis of partial differential equations. One tool commonly used in this setting, is modified equation analysis Warming & Hyett (1974), which determines how to better model discrete steps with a continuous differential equation by introducing higher order spatial or temporal derivatives. We present two methods based on modified equation analysis, which modify gradient flow to account for the effect of discretization.



(a) Modified Loss (b) Modified Flow

Figure 4: **Modeling discretization.** We visualize the trajectories of gradient descent and momentum (black dots), gradient flow with and without momentum (blue lines), and the modified dynamics (red lines) on the quadratic loss $\mathcal{L}(w) = w^\intercal \left[\begin{smallmatrix} 2.5 & -1.5 \\ -1.5 & 2 \end{smallmatrix}\right] w$. On the left we visualize gradient dynamics using modified loss. On the right we visualize momentum dynamics using modified flow. In both settings the modified continuous dynamics visually track the discrete dynamics better than the original continuous dynamics. See appendix C for further details.

Modified loss. Gradient descent always moves in the direction of steepest descent on a loss function $\mathcal{L}$ at each step, however, due to the finite nature of the learning rate, it fails to remain on the continuous steepest descent path given by gradient flow. Li et al. (2017); Feng et al. (2019) and most recently Barrett & Dherin (2020), demonstrate that the gradient descent trajectory closely follows the steepest descent path of a modified loss function $\widetilde{\mathcal{L}}$. The divergence between these trajectories fundamentally depends on the learning rate η and the curvature H. As derived in Barrett & Dherin (2020), and summarized in appendix C, this divergence is given by the gradient correction $-\frac{1}{2}Hg$, which is the gradient of the norm $-\frac{\eta}{4}\left|\nabla\mathcal{L}\right|$. Thus, the modified loss is $\widetilde{\mathcal{L}} = \mathcal{L} + \frac{\eta}{4}|\nabla\mathcal{L}|^2$ and the modified gradient flow ODE is

$$\frac{d\theta}{dt} = -g(\theta) - \frac{\eta}{2}H(\theta)g(\theta). \tag{16}$$

See Fig. 4 for an illustrative example of this method applied to a quadratic loss in $\mathbb{R}^2$.

Modified flow. Rather than modifying gradient flow with higher order "spatial" derivatives of the loss function, here we introduce higher order temporal derivatives. We start by assuming the existence of a continuous trajectory $\theta(t)$ that weaves through the discrete steps taken by gradient descent and then identify the differential equation that generates the trajectory. Rearranging the update equation for gradient descent, $\theta_{t+1} = \theta_t - \eta g(\theta_t)$, and assuming $\theta(t) = \theta_t$ and $\theta(t+\eta) = \theta_{t+1}$, gives the equality $-g(\theta_t) = \frac{\theta(t+\eta)-\theta(t)}{\eta}$, which Taylor expanding the right side results in the differential equation $-g(\theta_t) = \frac{d\theta}{dt} + \frac{\eta}{2}\frac{d^2\theta}{dt^2} + O(\eta^2)$. Notice that in the limit as $\eta \to 0$ we regain gradient flow. For small η, we obtain a modified version of gradient flow with an additional second-order term,

$$\frac{d\theta}{dt} = -g(\theta) - \frac{\eta}{2}\frac{d^2\theta}{dt^2}. \tag{17}$$

This approach to modifying first-order differential equation with higher order temporal derivatives was applied by Kovachki & Stuart (2019) to construct a more realistic continuous model for momentum, as illustrated in Fig. 4.

6 Combining Symmetry and Modified Gradient Flow to Derive Exact Learning Dynamics

As shown in section 4, each symmetry results in a conserved quantity under gradient flow. We now study how weight decay, momentum, stochastic gradients, and finite learning rates all interact to break these conservation laws. Remarkably, even when using a more realistic continuous model for stochastic gradient descent, as discussed in section 5, we can derive exact learning dynamics for the previously conserved quantities. To do this we (i) consider a realistic continuous model for SGD, (ii) project these learning dynamics onto the generator vector fields $\partial_\alpha \psi$ associated with each symmetry, (iii) harness the geometric constraints from section 3 to derive simplified ODEs, and (iv)

solve these ODEs to obtain exact dynamics for the previously conserved quantities. We first consider the continuous model of SGD without momentum incorporating weight decay, stochasticity, and modified loss. In this setting, the exact dynamics, as fully derived in appendix D, for the parameter combinations tied to the symmetries are,

$$\textbf{Translation:} \quad \langle \theta_{\mathcal{A}}(t), \mathbb{1} \rangle = e^{-\lambda t} \langle \theta_{\mathcal{A}}(0), \mathbb{1} \rangle \tag{18}$$

$$\textbf{Scale:} \quad |\theta_{\mathcal{A}}(t)|^2 = e^{-2\lambda t} |\theta_{\mathcal{A}}(0)|^2 + \eta \int_0^t e^{-2\lambda(t-\tau)} \left|g_{\mathcal{A}}\right|^2 d\tau \tag{19}$$

$$\textbf{Rescale:} \quad |\theta_{\mathcal{A}_1}(t)|^2 - |\theta_{\mathcal{A}_2}(t)|^2 = \tag{20}$$

$$e^{-2\lambda t}(|\theta_{\mathcal{A}_1}(0)|^2 - |\theta_{\mathcal{A}_2}(0)|^2) + \eta \int_0^t e^{-2\lambda(t-\tau)} \left(\left|g_{\theta_{\mathcal{A}_1}}\right|^2 - \left|g_{\theta_{\mathcal{A}_2}}\right|^2 \right) d\tau$$

Notice how these equations are equivalent to the conservation laws derived in section 4 when $\eta = \lambda = 0$. Remarkably, even in typical hyperparameter settings (weight decay, stochastic batches, finite learning rates), these solutions match nearly perfectly with empirical results from modern neural networks (VGG-16) trained on real-world datasets (Tiny ImageNet), as shown in Fig. 5. We will now discuss each equation individually.

Translation dynamics. For parameters with translation symmetry, equation 18 implies that the sum of these parameters ($\langle \theta_{\mathcal{A}}(t), \mathbb{1} \rangle$) decays exponentially to zero at a rate proportional to the weight decay. Equation 18 does not directly depend on the learning rate η nor any information of the dataset or task. This is due to the lack of curvature in the gradient field for these parameters (as shown in Fig. 2). This implies that at initialization we can deterministically predict the trajectory for the parameter sum as simple exponential functions with a rate defined by the weight decay. The first row in Fig. 5 demonstrates this qualitatively, as all trajectories are smooth exponential functions that converge faster for increasing levels of weight decay.

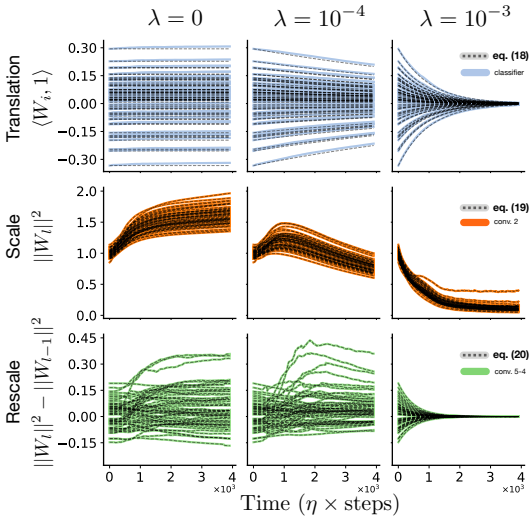


Figure 5: **Exact dynamics of VGG-16 on Tiny ImageNet.** We plot the column sum of the final linear layer (top row) and the difference between squared channel norms of the fifth and fourth convolutional layer (bottom row) of a VGG-16 model without batch normalization. We plot the squared channel norm of the second convolution layer (middle row) of a VGG-16 model with batch normalization. Both models are trained on Tiny ImageNet with SGD with learning rate $\eta = 0.1$, weight decay λ, batch size $S = 256$, for 100 epochs . Colored lines are empirical and black dashed lines are the theoretical predictions from equations (18), (19), and (20). See appendix H for more details on the experiments and how we compute the integrals terms in the exact solutions.

Scale dynamics. For parameters with scale symmetry, equation 19 implies that the norm for these parameters ($|\theta_{\mathcal{A}}|^2$) is the sum of an exponentially decaying memory of the norm at initialization and an exponentially weighted integral of gradient norms accumulated through training. Compared to the translation dynamics, the scale dynamics do depend on the data through the gradient norms accumulated throughout training. Without weight decay $\lambda = 0$, the first term stays constant and the second term grows monotonically. With weight decay $\lambda > 0$, the first term decays monotonically to zero, while the second term can decay or grow, but always stays positive. The second row in Fig. 5 demonstrates these qualitative relationships. Without weight decay the norms increases monotonically as predicted and with weight decay the dynamics are non-monotonic and present more complex behavior. To better understand the forces driving these complex dynamics, we can examine the time derivative of equation 19,

$$\frac{d}{dt} |\theta_{\mathcal{A}}(t)|^2 = -2\lambda |\theta_{\mathcal{A}}(t)|^2 + \eta \left|g_{\mathcal{A}}\right|^2 . \tag{21}$$

From this equation we see that there is a competition between a centripetal effect due to weight decay ($-2\lambda|\theta_{\mathcal{A}}(t)|^2$) and a centrifugal effect due to discretization ($\eta\,|g_{\mathcal{A}}|^2$). The centripetal effect due to weight decay is a direct consequence of its regularizing influence, pulling the parameters towards the origin. The centrifugal effect due to discretization originates from the spherical geometry of the gradient field in parameter space – because scale symmetry implies the gradient is always orthogonal to the parameter itself, each discrete update with a finite learning rate effectively pushes the parameters away from the origin. At the stationary state of the dynamics, these forces will balance leading the dynamics of these parameters to be constrained to the surface of a high-dimensional sphere. In particular, at stationarity, then $\frac{d}{dt}|\theta(t)|^2 = 0$, which rearranging equation (21) gives the condition $\omega(t) \equiv \left|\frac{d\theta}{dt}\right| / |\theta| = \sqrt{\frac{2\lambda}{\eta}}$. Consistent with the results of Wan et al. (2020), this implies that at stationarity the angular speed $\omega(t)$ of the weights is constant and governed only by the learning rate η and weight decay constant λ.

Rescale dynamics. For parameters with rescale symmetry, equation 20 is the sum of an exponentially decaying memory of the difference in norms at initialization and an exponentially weighted integral of difference in gradient norms accumulated through training. Similar to the scale dynamics, the rescale dynamics do depend on the data through the gradient norms, however unlike the scale dynamics we have no guarantee that the integral term is always positive. This leads to quite sophisticated, complex dynamics, consistent with the third row in Fig. 5. Despite the complexity, our theory, nevertheless, quantitatively matches the empirics. The only apparent pattern from the empirics is that for large enough weight decay, the regularization dominates any complexity introduced by the gradient norms and the difference in parameter norms decays exponentially to zero.

Harmonic oscillation with momentum. We will now consider a continuous model of SGD *with* momentum. As discussed in appendix E, we consider the continuous model incorporating weight decay, momentum, stochasticity, and modified flow. Under this model, the solutions we obtain take the form of driven harmonic oscillators where the driving force is given by the gradient norms, the friction is defined by the momentum constant, the spring coefficient is defined by the regularization rate, and the mass is defined by the the learning rate and momentum constant. For most standard hyperparameter choices, these solutions are in the overdamped setting and align well with the first-order solutions for SGD without momentum up to a time rescaling, as shown in the left and middle panel of Fig. 6. However, for large values of beta we can push the solution into the underdamped regime where we would expect harmonic oscillation and indeed, we can empirically verify our predictions, even at scale for VGG-16 trained on Tiny ImageNet, as in right panel of Fig. 6.

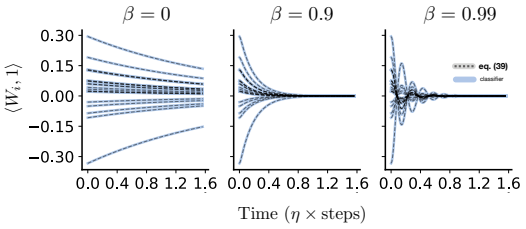


Figure 6: **Momentum leads to harmonic oscillation.** We plot the column sum of the final linear layer of a VGG-16 model (without batch normalization) trained on Tiny ImageNet with SGD with learning rate $\eta = 0.1$, weight decay $\lambda = 5 \times 10^{-3}$, batch size $S = 256$ and momentum $\beta \in \{0, 0.9, 0.99\}$. Colored lines are empirical and black dashed lines are the theoretical predictions from equations (39).

7 Conclusion

Despite being the central guiding principle in the exploration of the physical world Anderson (1972); Gross (1996), symmetry has been underutilized in understanding the mechanics of neural networks. In this paper, we constructed a unifying theoretical framework harnessing the geometric properties of symmetry and realistic continuous equations for learning dynamics modeling weight decay, momentum, stochasticity, and discretization. We use this framework to derive *exact* dynamics for meaningful combinations of parameters, which we experimentally verified on large scale neural networks and datasets. For example, in the case of a VGG-16 model with batch normalization trained on Tiny-ImageNet (one of the model/dataset combinations we considered in section 6) there are $12,751$ distinct parameter combinations whose dynamics we can analytically describe. Overall, this work provides the first solid step towards the mechanics of learning in neural networks.

REFERENCES

Jing An, Jianfeng Lu, and Lexing Ying. Stochastic modified equations for the asynchronous stochastic gradient descent. *Information and Inference: A Journal of the IMA*, 2018.

Philip W Anderson. More is different. *Science*, 177(4047):393–396, 1972.

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018a.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018b.

Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. *arXiv preprint arXiv:1812.03981*, 2018c.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pp. 8141–8150, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

David GT Barrett and Benoit Dherin. Implicit gradient regularization. *arXiv preprint arXiv:2009.11162*, 2020.

Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–10. IEEE, 2018.

Vitaliy Chiley, Ilya Sharapov, Atli Kosson, Urs Koster, Ryan Reece, Sofia Samaniego de la Fuente, Vishal Subbiah, and Michael James. Online normalization for training neural networks. In *Advances in Neural Information Processing Systems*, pp. 8433–8443, 2019.

Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. In *Advances in Neural Information Processing Systems*, pp. 5225–5235, 2017.

Simon S Du and Wei Hu. Width provably matters in optimization for deep linear neural networks. *arXiv preprint arXiv:1901.08572*, 2019.

Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, pp. 384–395, 2018a.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018b.

Yuanyuan Feng, Tingran Gao, Lei Li, Jian-Guo Liu, and Yulong Lu. Uniform-in-time weak error analysis for stochastic gradient descent algorithms via diffusion approximation. *arXiv preprint arXiv:1902.00635*, 2019.

David J Gross. The role of symmetry in fundamental physics. *Proceedings of the National Academy of Sciences*, 93(25):14256–14259, 1996.

Elad Hoffer, Ron Banner, Itay Golan, and Daniel Soudry. Norm matters: efficient and accurate normalization schemes in deep networks. In *Advances in Neural Information Processing Systems*, pp. 2160–2170, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in neural information processing systems*, pp. 586–594, 2016.

Nikola B Kovachki and Andrew M Stuart. Analysis of momentum methods. *arXiv preprint arXiv:1906.04285*, 2019.

Daniel Kunin, Jonathan M Bloom, Aleksandrina Goeva, and Cotton Seed. Loss landscapes of regularized linear autoencoders. *arXiv preprint arXiv:1901.08168*, 2019.

Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pp. 8572–8583, 2019.

Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pp. 2101–2110, 2017.

Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*, 2019.

Zhiyuan Li, Kaifeng Lyu, and Sanjeev Arora. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate. *Advances in Neural Information Processing Systems*, 33, 2020.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 888–896, 2019.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Continuous-time limit of stochastic gradient descent revisited. *NIPS-2015*, 2015.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research*, 18(1):4873–4907, 2017.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Behnam Neyshabur, Russ R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2015.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

David Saad and Sara Solla. Dynamics of on-line gradient descent learning for multilayer neural networks. *Advances in neural information processing systems*, 8:302–308, 1995.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems*, pp. 901–909, 2016.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Samuel L Smith and Quoc V Le. A bayesian perspective on generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451*, 2017.

Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in neural information processing systems*, pp. 2510–2518, 2014.

Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *arXiv preprint arXiv:2006.05467*, 2020.

Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics of deep neural networks with batch normalization and weight decay. *arXiv preprint arXiv:2006.08419*, 2020.

RF Warming and BJ Hyett. The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of computational physics*, 14(2):159–179, 1974.

Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:1810.00004*, 2018.

Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *arXiv preprint arXiv:1803.00195*, 2018.