Noether's Learning Dynamics: Role of Symmetry Breaking in Neural Networks

Hidenori Tanaka[1*] , **Daniel Kunin**[2]
[1]Physics & Informatics Laboratories, NTT Research, Inc., Sunnyvale, CA, USA
[2]Stanford University, Stanford, CA, USA

Abstract

In nature, symmetry governs regularities, while symmetry breaking brings texture. In artificial neural networks, symmetry has been a central design principle to efficiently capture regularities in the world, but the role of symmetry breaking is not well understood. Here, we develop a theoretical framework to study the *geometry of learning dynamics* in neural networks, and reveal a key mechanism of explicit symmetry breaking behind the efficiency and stability of modern neural networks. To build this understanding, we model the discrete learning dynamics of gradient descent using a continuous-time Lagrangian formulation, in which the learning rule corresponds to the kinetic energy and the loss function corresponds to the potential energy. Then, we identify *kinetic symmetry breaking* (KSB), the condition when the kinetic energy explicitly breaks the symmetry of the potential function. We generalize Noether's theorem known in physics to take into account KSB and derive the resulting motion of the Noether charge: *Noether's Learning Dynamics* (NLD). Finally, we apply NLD to neural networks with normalization layers and reveal how KSB introduces a mechanism of *implicit adaptive optimization*, establishing an analogy between learning dynamics induced by normalization layers and RMSProp. Overall, through the lens of Lagrangian mechanics, we have established a theoretical foundation to discover geometric design principles for the learning dynamics of neural networks.

With the rapid increase in available data and computational power, artificial neural networks have become an essential tool both in science and engineering. To obtain an accurate model, one has to overcome the challenge of tuning millions of randomly initialized parameters $q \in \mathbb{R}^N$ to minimize the loss function $f(q) \in \mathbb{R}$, a measure of the discrepancy between the model's predictions from the input data and the provided truth. In minimizing the loss, the learning rule specifies how to iteratively update these parameters based on the recent trajectory and the local geometry of the non-convex loss landscape. We can represent the repeated updates of the millions of parameters during learning as discrete movements of a point in high-dimensional parameter space (Fig. 1a). The architecture shapes the static loss landscape and the learning rule governs the motion of the model through parameter space during training. However, it is the complex interaction of the architecture and learning rule which determines the final state of the trained model. Inspired by tools and ideas from classical mechanics (Fig. 1b), we develop a theoretical framework which unifies the effects of the architecture and learning rule on the *geometry of the learning dynamics*.

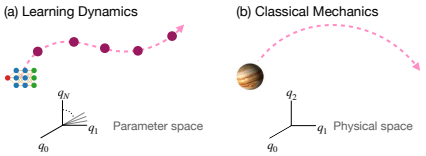

Figure 1: **Learning dynamics in analogy to classical mechanics.** (a) Learning dynamics of a neural network in parameter space. (b) Motion of a particle in physical space.

* Correspondence to: Hidenori Tanaka <hidenori.tanaka@ntt-research.com>
[2] Work done during an internship at Physics & Informatics Laboratories, NTT Research, Inc.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Symmetry as a mathematical lens. One of the most powerful mathematical tools for studying the geometry of complex objects is symmetry. A symmetry of an object is a transformation that when applied to the object leaves certain properties unchanged. For example, consider how a circle can be rotated around its center without changing its overall appearance. In otherwords, a circle has a continuous rotational symmetry. In particular, the form of the symmetry transformation constrains the form of the object, the circle is the only 2D shape that has continuous rotational symmetry. Objects with symmetry are not restricted to geometric shapes, but could be functions or even dynamical systems. Recently, the lens of symmetry has been applied in the study of neural networks [1, 2]. A recent work [1] identified an array of symmetries in neural networks to study the geometry of the loss function $f(q)$ *in the parameter space*. For example, batch normalization [3] defined as $\mathrm{BN}(qx) = (qx - \mathrm{E}[qx])/\sqrt{\mathrm{Var}[qx]}$ introduces a scale symmetry in the parameters immediately preceding the normalization layer, $\mathrm{BN}((1+s)qx) = \mathrm{BN}(qx)$ for any positive scalar $s \in \mathbb{R}^+$.

Geometry of learning dynamics. What does the symmetry of the loss function in parameter space imply about the symmetry of the learning dynamics? In classical mechanics, a symmetry of the potential function in space usually implies the *equivariance* of the trajectory, namely, a solution of the Newton's equation of motion (grey) remains a solution under the symmetry transformations (red) as shown in Fig. 2 (a). This implies that the time evolution of the potential energy is *invariant* under symmetry transformations at any point in the dynamics. However, this intuition does not hold in modern neural networks. Consider the concrete example of a VGG11 model with a batch normalization layer after each convolutional filter. The scale symmetry of batch normalization implies the loss function is invariant to the scaling transformation on the convolutional filters q, $f((1+s)qx) = f(qx)$, however the learning dynamics are not. To demonstrate this we train two VGG11 models on Tiny ImageNet with a standard initialization (grey) and a scaled initialization (red). Notice, that the scaling does not change the initial loss due to the scale symmetry, however it does effect the *learning dynamics* of the loss, as shown in the bottom of Fig. 2(b). In otherwords, even though the loss landscape observes scale symmetry at all steps in training, the learning dynamics do not. Understanding symmetry of the learning dynamics requires a new theoretical perspective.

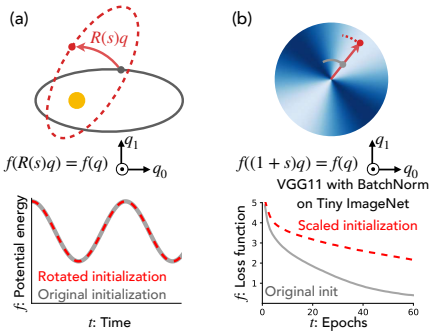

Figure 2: **Symmetry of the loss $\neq$ symmetry of the learning dynamics.** (a) Physical dynamics with rotational symmetry. (b) Learning dynamics in a parameter space with scale symmetry.

Implicit acceleration and variational perspective on learning dynamics. In physics, variational formulations based on the Lagrangian and Hamiltonian provide a unifying framework to study the role of symmetry in dynamical systems. However, there is still a gap between the tools and concepts in physics and the learning dynamics of neural networks. We identify the root cause of this gap to be the learning rule, namely the fact that the continuous-time limit of gradient descent is a first order differential equation in time, gradient flow $\frac{dq}{dt} = -\nabla f$, unlike classical mechanics which is governed by Netwon's second law $m\frac{d^2q}{dt^2} = -\nabla f$, which is second-order in time. We bridge this gap by realizing that the effect of discretization, is to introduce an *implicit acceleration* term proportional to the learning rate. Leveraging this connection, we build the *mechanics of learning dynamics*:

Sec. 1. From Newtonian to Lagrangian mechanics of learning via *implicit acceleration*. We show how gradient descent with a finite learning rate introduces *implicit acceleration* making it amendable to the Lagrangian formulation of accelerated optimizers [4].

Sec. 2. Unlike in physics, *kinetic symmetry breaking* (KSB) is inherent in learning dynamics. We define *kinetic symmetry breaking*, where the kinetic energy corresponding to the learning rule explicitly breaks the symmetry of the potential energy corresponding to the loss function.

Sec. 3. We derive *Noether's learning dynamics* (NLD) induced by the KSB. We generalize Noether's theorem in physics to derive NLD, which accounts for damping, the unique symmetries of the loss, and the non-Euclidean metric used in learning rules.

Sec. 4. KSB induces *impicit adaptive optimization* and enables successful learning. We leverage NLD to exactly solve for the time-evolution of the effective learning rate for networks with normalization layers, and establish an exact analogy between two seemingly unrelated components of modern deep learning: normalization and adaptive optimization.

1 From Newtonian to Lagrangian mechanics of learning

The fundamental law of mechanics is the principle of least action [5]. We first review how pioneering works by Wibisono, Wilson, and Jordan [4, 6, 7] apply the variational principles of physics to model accelerated methods in optimization. We then show how gradient descent with a finite learning rate introduces *implicit acceleration* making it amendable to the Lagrangian formulation.

Variational formulation of classical mechanics. Just over a century after Newton, Joseph-Louis Lagrange demonstrated how Newton's equations of motion could be derived from a variational framework of least action. Rather than integrating an equation of motion over time to get the path an object takes from point a to point b, Lagrange prescribed that the path taken by the object is the path that minimizes the action, the path integral of a function termed the Lagrangian, over all possible paths from a to b. This principle of least action transformed mechanics from a differential description to a variational description. Consider for example the simple setup of a particle of mass m at position $q \in \mathbb{R}^N$ in N dimensional physical space moving according to a potential $f(q) \in \mathbb{R}$ and Newton's second law $m\ddot{q} = -\nabla f$, where $\ddot{q} = \frac{d^2q}{dt^2}$ is the acceleration. Let the Lagrangian be the function $\mathcal{L}(q, \dot{q}, t) = \frac{1}{2}m\dot{q}^2 - f(q)$, the difference between the kinetic energy $T = \frac{1}{2}m\dot{q}^2$ and the potential energy $V = f(q)$ of the particle. The principle of least action states that the time evolution of the particle's position $q(t)$ is the function that minimizes the action $S[q] = \int_{t_0}^{t_1} \mathcal{L}(q, \dot{q}, t)dt$. In otherwords, $q(t)$, with boundary conditions $q(t_0) = a$ and $q(t_1) = b$, must satisfy the stationary condition $\frac{\delta S}{\delta q(t)} = 0$, which is equivalent to satisfying the Euler-Lagrange equation (see SI Sec. A for the review),

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) = \frac{\partial \mathcal{L}}{\partial q}. \tag{1}$$

Indeed, substituting the Lagrangian into the Euler-Lagrange equation, we recover the Newton's equation of motion $m\ddot{q} = -\nabla f$ confirming the consistency of this variational formulation.

Damping in optimization. In learning dynamics, one of the universal features is damping, a term proportional to the velocity $\dot{q} = \frac{dq}{dt}$ in the equation motion. Intuitively, damping is needed for any optimization, since without damping the learning dynamics would conserve energy and not converge. We can incorporate damping into the previous Lagrangian example by introducing an explicit time-dependence $\mathcal{L}(q, \dot{q}, t) = e^{\frac{\mu}{m}t}(\frac{1}{2}m\dot{q}^2 - f(q))$, where μ is the damping (or friction) coefficient [8]. Substituting this Lagrangian into the Euler-Lagrange equation and dividing both sides by $e^{\frac{\mu}{m}t}$, we obtain the equation of motion of a damped particle $m\ddot{q} + \mu\dot{q} = -\nabla f$ as intended.

Non-Euclidean geometry in optimization. Many optimization algorithms assume an underlying geometry. For example, consider gradient descent with learning rate η. The update equation, $q(t + \eta) = q(t) - \eta\nabla f$, is equivalent to solving the minimization problem, $q(t + \eta) = \arg\min_{q'} \left[\langle \nabla f(q(t)), q' \rangle + \eta^{-1}\frac{1}{2}|q' - q(t)|^2\right]$, where $\langle\ ,\ \rangle$ represents inner product. In otherwords, we can interpret each update of gradient descent as a trade-off (specified by η) between minimizing the loss $f(q)$, while remaining close to its current location in Euclidean distance. For non-Euclidean measures of distance we would obtain different optimization algorithms. Mirror descent generalizes this idea by replacing the distance measure by the Bergman divergence $D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$ where $h(x)$ is a distance-generating function. Indeed, when the distance generating function is $h(x) = \frac{1}{2}|x|^2$, then the Bregman divergence simplifies to be the squared Euclidean distance $D_h(y, x) = \frac{1}{2}|x - y|^2$ and mirror descent is equivalent to gradient descent. The mirror descent framework encapsulates a broad array of optimization methods such as gradient descent, natural gradient descent, and others [9].

	α_t	β_t	γ_t	$h(x)$	Euler-Lagrange equation
Nesterov momentum	$\log 2 - \log t$	$2\log t + \log 1/4$	$2\log t$	$\frac{1}{2}\lvert x\rvert^2$	$\ddot{q} + 3/t + \nabla f = 0$
Damped physical motion	$-\log m$	$\log m$	$\frac{\mu}{m}t$	$\frac{1}{2}\lvert x\rvert^2$	$m\ddot{q} + \mu\dot{q} + \nabla f = 0$
Natural gradient flow	$-\log m$	$\log m$	$\frac{\mu}{m}t$	$h(x)$	$\dot{q} + F^{-1}\nabla f = 0\ (m \to 0)$
GD + momentum	$-\log\frac{\eta(1+\beta)}{2}$	$\log\frac{\eta(1+\beta)}{2}$	$\frac{2(1-\beta)}{\eta(1+\beta)}t$	$\frac{1}{2}\lvert x\rvert^2$	$\frac{\eta}{2}(1+\beta)\ddot{q} + (1-\beta)\dot{q} + \nabla f = 0$

Table 1: The Bregman Lagrangian [4, 6] provides a unified description of a family of learning rules with time dependent parameters α_t, β_t, γ_t and metric $h(x)$ which provide the algorithmic degrees of freedom.

The Bregman Lagrangian. Integrating building blocks of optimization such as damping and non-Euclidean geometry into the variational framework of mechanics, Wibisono, Wilson, and Jordan introduced the Bregman Lagrangian [4, 6],

$$\mathcal{L}(q, \dot{q}, t) = e^{\alpha_t + \gamma_t} \left(D_h(q + e^{-\alpha_t}\dot{q}, q) - e^{\beta_t} f(q) \right). \tag{2}$$

For various choices of the time-dependent factors α_t, β_t, and γ_t the Bregman Lagrangian, under the principle of least action, defines continuous approximations to an array of learning rules such as Nesterov's accelerated gradient methods [10, 11] and natural gradient flow [9]. The flexibility of this framework makes our analysis generalize to a broad array of learning rules, as summarized in table 1.

Implicit acceleration and the Lagrangian formulation of gradient descent. In the continuous-time limit of diminishing learning rate $\eta \to 0$, gradient descent $q(t+\eta) = q(t) - \eta \nabla f$ becomes a first order differential equation, gradient flow $\dot{q} = -\nabla f$. Unlike Nesterov's momentum [11], the trajectory of gradient descent with heavy ball momentum β as applied in deep learning [12], follows gradient flow with the rescaling $(1-\beta)\dot{q} = -\nabla f$. Here, we apply modified equation analysis [1, 13] to incorporate the effect of a finite learning rate and leverage the fact that the discrete steps of gradient descent closely follow the continuous-time trajectory of the second-order differential equation in time. We can informally see this by Taylor expanding an update with respect to a small, but finite learning rate $q(t+\eta) = q + \eta\dot{q} + \frac{\eta^2}{2}\ddot{q} + O(\eta^3)$ and then plugging this expression to the gradient descent updates $\frac{q(t+\eta)-q(t)}{\eta} = -\nabla f$. We keep the first order in η and obtain a second-order differential equation $\frac{\eta}{2}\ddot{q} + \dot{q} = -\nabla f$. We call the additional term $\frac{\eta}{2}\ddot{q}$, *implicit acceleration*. This dynamics reduces to gradient flow $\dot{q} = -\nabla f$ in the continuous-time limit as expected. (Note that a complementary approach is to view the effects of discrete step size as *implicit gradient regularization* on the original loss $f(q) \to f(q) + \frac{\eta}{4}|\nabla f|^2$ [1, 14, 15, 16, 17].) Performing a similar analysis [1, 13] accounting for the effects of the heavy ball momentum with weight decay k yields a second-order differential equation,

$$\frac{\eta}{2}(1+\beta)\ddot{q} + (1-\beta)\dot{q} + \nabla f(q) + kq = 0, \tag{3}$$

where $\frac{\eta}{2}(1+\beta)\ddot{q}$ represents the implicit acceleration.

Indeed, we can connect this second-order dynamics to the Bregman Lagrangian by considering the factors $\alpha_t = -\log\frac{\eta(1+\beta)}{2}$, $\beta_t = \log\frac{\eta(1+\beta)}{2}$, $\gamma_t = \frac{2(1-\beta)}{\eta(1+\beta)}t$, and the distance generating function $h(x) = \frac{1}{2}|x|^2$. The Lagrangian describing the learning dynamics under gradient descent with a finite learning rate η, heavy ball momentum β, the loss function $f(q)$, and weight decay constant k is

$$\mathcal{L}(q, \dot{q}, t) = e^{\frac{2(1-\beta)}{\eta(1+\beta)}t} \left[\frac{\eta(1+\beta)}{4}|\dot{q}|^2 - \left(f(q) + \frac{k}{2}|q|^2 \right) \right]. \tag{4}$$

2 Symmetry and symmetry breaking in neural networks

The variational perspective of optimization, discussed in the previous section, provides a useful lens to characterizing the path a model takes through parameter space during training. To study the geometry of this path we will turn to the tools of symmetry. In particular, in this section we will discuss not only when symmetries exist in the loss function, but also when they are broken by the learning rule.

2.1 Symmetries of the loss function

There have been a number of works studying symmetries of the loss function in parameter space, such as [18, 19] who study discrete permutation symmetries or [1] who study continuous differentiable symmetries. In this work we will focus on continuous differentiable symmetries found in modern deep learning architectures. Consider a one-parameter family of differentiable maps $q \to Q(q, s)$ parameterized by a scalar $s \in \mathbb{R}$ such that $s = 0$ gives an identity $Q(q, 0) = q$. We say the potential $f(q)$ possesses a differentiable symmetry if its invariant to the transformation $f(Q(q, s)) = f(q)$ for all s. The symmetries we consider in this work are: *translation symmetry* where $Q(q, s) = q + s\hat{n}$ and $\hat{n}$ is a time-independent vector, *rotation symmetry* where $Q(q, s) = R(s)q$ and $R(s)$ is a rotation matrix, and *scale symmetry* where $Q(q, s) = (1+s)q$. Translation symmetries occur in a neural network for parameters immediately preceding a softmax function. Rotation symmetry is one of the

most fundamental symmetry groups in classical mechanics and occur in neural networks with weight decay [20, 21] or word embeddings [22]. Scale symmetries appear ubiquitously in neural networks for parameters immediately preceding normalization layers, such as batch normalization [3] and its variants [23]. There is also a related symmetry, defined as rescale symmetry in [1], present in neural networks with continuous, homogeneous activation functions (e.g. ReLU, Leaky ReLU, linear).

2.2 Symmetry breaking in the kinetic energy

In the previous section, we introduced the Lagrangian formulation of learning dynamics, in which the learning rule corresponds to the kinetic energy and the loss function corresponds to the potential energy. The kinetic energy of the Bregman Lagrangian, defined as

$$T_h(q, \dot{q}, t) \equiv e^{\alpha_t} D_h(q + e^{-\alpha_t}\dot{q}, q) = e^{\alpha_t} \left(h(q + e^{-\alpha_t}\dot{q}) - h(q)\rangle - \langle \nabla h(q), e^{-\alpha_t}\dot{q}\rangle \right), \quad (5)$$

is as important as the loss function, but its properties are less discussed. Here, we ask if the Bregman kinetic energy respects the same symmetries as the loss function, whether $T_h(Q(q,s), \dot{Q}(q,s), t) = T_h(q, \dot{q}, t)$.

Euclidean kinetic energy. First, we consider when $h(x) = \frac{1}{2}|x|^2$ and study the symmetry properties of the Euclidean kinetic energy,

$$T_{\mathrm{E}}(q, \dot{q}, t) = \frac{e^{-\alpha_t}}{2}|\dot{q}|^2. \quad (6)$$

For translation symmetry the velocity is invariant under the transformation $\dot{Q} = \dot{q}$ and thus indicating that the Euclidean kinetic energy is invariant under spatial translation. For rotation symmetry, the temporal differentiation commutes with the rotation $\dot{Q} = R\dot{q}$, however the Euclidean kinetic energy is invariant under spatial rotation because the norm is unchanged $|\dot{Q}|^2 = \langle R\dot{q}, R\dot{q}\rangle = |\dot{q}|^2$. However, for scale symmetry $\dot{Q} = (1+s)\dot{q}$ and thus the norm $|\dot{Q}|^2 = (1+s)^2|\dot{q}|^2$ explicitly depends on s. Therefore, the kinetic energy explicitly breaks the scale symmetry present in the loss function. We call this explicit symmetry breaking of the Lagrangian by the kinetic energy term *Kinetic Symmetry Breaking* (KSB).

Non-Euclidean kinetic energy. Next, we consider the setting of a general non-Euclidean metric. The Taylor expansion of the Bregman kinetic energy with respect to the first term $h(q + e^{-\alpha_t}\dot{q})$ in Eq.5 results in the simplification,

$$T_h(q, \dot{q}, t) = (e^{-\alpha_t}/2) \left\langle \dot{q}, \nabla^2 h(q)\dot{q} \right\rangle + O\left((e^{-\alpha_t})^2 \right). \quad (7)$$

Notice, the Euclidean kinetic energy only depends on the squared norm of the velocity $|\dot{Q}|^2$ because the Hessian is the identity matrix $\nabla^2 \frac{1}{2}|x|^2 = I$ and the higher order derivatives are zero. With non-Euclidean metrics, such simplifications special to the Euclidean case do not necessarily apply and we should not expect that the Bregman kinetic energy respects any symmetries in general. However, the natural gradient [9] and its elegant approximations, such as [24], are examples of non-Euclidean optimization methods, which have invariance properties by construction. While the invariance properties of the natural gradient descent can be broken due to discretization [25], the fast and accurate approximation of natural gradients in modern large-scale neural networks is an active area of research [26, 27].

3 Noether's learning dynamics

So far, we have seen how the Bregman Lagrangian provides a unified variational description of learning dynamics, and the scale symmetry, ubiquitous in modern neural networks, induces kinetic symmetry breaking (KSB). Here, we apply Noether's theorem to the Bregman Lagrangian and derive *Noether's Learning Dynamics* (NLD), a unified equality that holds for any combination of symmetry and learning rules.

Noether's theorem. In 1918 Emmy Noether famously described one of the most fundamental theorems in modern physics: every differentiable symmetry of the action has an associated conserved quantity through time [28]. Noether's theorem not only proves the existence of, but also provides an explicit expression for, the conserved quantity, often called Noether charge. Focusing just on the symmetry properties of the Lagrangian, a general form of Noether's theorem relates the dynamics of

the Noether charge $\langle \partial_{\dot{q}}\mathcal{L}, \partial_s Q\rangle$ to the change in the Lagrangian under an infinitesimal transformation $\partial_s \mathcal{L}$ as

$$\frac{d}{dt}\langle \partial_{\dot{q}}\mathcal{L}, \partial_s Q\rangle = \partial_s \mathcal{L}, \tag{8}$$

where we evaluate all derivatives at the identity element of the symmetry group ($s = 0$). We can derive this relationship as $\partial_s \mathcal{L} = \langle \partial_q \mathcal{L}, \partial_s Q\rangle + \langle \partial_{\dot{q}}\mathcal{L}, \partial_s \dot{Q}\rangle = \frac{d}{dt}\langle \partial_{\dot{q}}\mathcal{L}, \partial_s Q\rangle$, where we plugged in the Euler-Lagrange equation $\partial_q \mathcal{L} = \frac{d}{dt}(\partial_{\dot{q}}\mathcal{L})$ and rearranged the total time derivative. The most common application of Noether's theorem in physics is when the Lagrangian, including both kinetic and potential energies, is symmetric under the infinitesimal transformation, $\partial_s \mathcal{L} = 0$, implying Noether charge is conserved through time. For example, if we assume symmetry of the Lagrangian due to homogeneity or isotropy of space, Noether's theorem directly implies the conservation of momentum or angular momentum respectively. However, as shown in Sec. 2, the Bregman kinetic energy for learning systems is often asymmetric to the symmetry of the potential function, implying that the resulting Noether charge is *not* conserved.

Noether's learning dynamics. Here, we derive a general equation describing the kinetic-asymmetry-induced dynamics for the Noether charge in learning systems. If the Bregman kinetic energy does not respect the symmetry of the loss function, then Eq. 8 becomes

$$\frac{d}{dt}\langle \partial_{\dot{q}}\mathcal{L}, \partial_s Q\rangle = e^{\gamma_t}\partial_s T_h, \tag{9}$$

implying kinetic symmetry breaking ($\partial_s T_h \neq 0$) induces motion of the Noether charge. To build better intuition for this motion and inspired by the definition of the generalized momentum, we define the *Bregman momentum* as,

$$\Delta_h \equiv e^{-\gamma_t}\partial_{\dot{q}}\mathcal{L} = \nabla h(q + e^{-\alpha_t}\dot{q}) - \nabla h(q). \tag{10}$$

Expanding Eq. 9 by plugging in the definition of the Bregman Lagrangian Eq. 2, the Bregman kinetic energy Eq. 5, and the Bregman momentum Eq 24 (details are described in SI Sec. B), we obtain *Noether's Learning Dynamics* (NLD),

$$\frac{d}{dt}\overbrace{\langle \Delta_h, \partial_s Q\rangle}^{\text{charge}} + \overbrace{\dot{\gamma}_t\langle \Delta_h, \partial_s Q\rangle}^{\text{damping}} = \overbrace{\langle \Delta_h, \partial_s \dot{Q}\rangle}^{\text{kinetic asymmetry}} + \overbrace{e^{\alpha_t}\langle \Delta_h - e^{-\alpha_t}\nabla^2 h(q)\dot{q}, \partial_s Q\rangle}^{\text{non-Euclidean metric}}. \tag{11}$$

Each term in NLD has an intuitive meaning. Recall the conventional setting in classical mechanics, where the entire Lagrangian is symmetric, there is no damping, and the metric is Euclidean. In this case, the Noether charge is the inner product of momentum and the generator of the symmetry transformation $\langle e^{-\alpha_t}\dot{q}, \partial_s Q\rangle$. The first term of NLD, $\langle \Delta_h, \partial_s Q\rangle$, is the inner product of the Bregman momentum and the generator of the symmetry transformation. In the setting of no damping, this terms is a generalization of the conventional Noether charge that encompasses non-Euclidean metrics. Indeed, in the Euclidean case, we can see that $\Delta_E = e^{-\alpha_t}\dot{q}$ and the term reduces to the conventional Noether charge as expected. The second term of NLD, $\dot{\gamma}_t\langle \Delta_h, \partial_s Q\rangle$, represents the contribution of damping and the term becomes zero when γ_t is constant and there is no damping. The third term of NLD, $\langle \Delta_h, \partial_s \dot{Q}\rangle$, represents the contribution of kinetic symmetry breaking. Under the Euclidean metric, this term is proportional to $\langle \dot{q}, \partial_s \dot{Q}\rangle$. For translation symmetry, $\partial_s \dot{Q} = 0$ and this term is zero $\langle \Delta_h, 0\rangle = 0$. Similarly, for rotation symmetry, this term is zero $\langle R|_{s=0}\dot{q}, (\partial_s R)\dot{q}\rangle = \langle \dot{q}, (\partial_s R)\dot{q}\rangle = 0$. Here, we used the fact that the generator of rotation is a skew-symmetric matrix. However, for scale symmetry, $\partial_s \dot{Q} = \dot{q}$ unique to learning systems, this term is proportional to $|\dot{q}|^2$ and thus present. The fourth term of NLD, $e^{\alpha_t}\langle \Delta_h - e^{-\alpha_t}\nabla^2 h(q)\dot{q}, \partial_s Q\rangle$, represents the contribution of the non-Euclidean geometry and is zero under the Euclidean metric $h(x) = \frac{1}{2}|x|^2$.

Noether's learning dynamics under gradient flow. Next, we show that in the over-damped limit of NLD, the charge $\langle \Delta_h, \partial_s Q\rangle$ itself, rather than its time-derivative becomes zero. Consider Eq.11 when $\alpha_t = -\log m$, $\beta_t = \log m$, $\gamma_t = \frac{\mu}{m}t$,

$$m\frac{d}{dt}\langle \Delta_h, \partial_s Q\rangle + \mu\langle \Delta_h, \partial_s Q\rangle = m\partial_s T. \tag{12}$$

When $\partial_s T = 0$, this equation describes how the charge decays exponentially with damping $\langle \Delta_h, \partial_s Q\rangle = e^{-\frac{\mu}{m}t}$ as in Fig. 3. In the mass-less limit $m \to 0$ with a fixed friction μ, the terms proportional to m in Eq. 12 all vanish and we get

$$\langle \Delta_h, \partial_s Q\rangle = 0, \tag{13}$$

which states that the charge itself is zero, rather than its time derivative. Furthermore, in the Euclidean settings, the result simplifies to $\langle \dot{q}, \partial_s Q \rangle = 0$. As a physical analogy, this means that quantities such as momentum or angular momentum becomes zero in the over-damped limit. This formula unifies an array of conservation properties noticed under gradient flow [1, 29, 30, 31, 32] with a formal theoretical connection to Noether's theorem.

Overall, just like how Noether's theorem [28] unified an array of conservation laws and provided a theoretical foundation to discover new ones in physics, we have further unified conservation laws previously observed in learning systems [1, 29, 30, 31, 32] and generalized these results for any combination of differentiable symmetries in neural network architectures and learning rules (e.g., natural gradient descent, Nesterov's accelerated gradient descent, accelerated mirror descent, and cubic-regularized Newton's method).

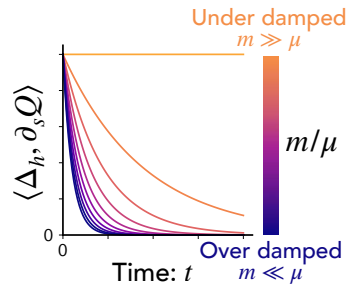

Figure 3: **Noether's theorem with damping.** Without KSB, the charge decays exponentially, but in the under-damped limit $m/\mu \gg 1$, the charge becomes a conserved quantity. In the over-damped limit $m \to 0$, the charge exponentially diminishes to zero.

4 Kinetic symmetry breaking induces implicit adaptive optimization

In the previous sections, we identified the mechanism of kinetic symmetry breaking and derived the Noether's learning dynamics (NLD). Here, we explore a role of symmetry breaking in neural networks by applying NLD to study auto rate-tuning by normalization layers, originally studied by Arora, Li, and Lyu [33]. In particular, we establish an exact theoretical analogy between two seemingly unrelated key components of modern deep learning: normalization and adaptive optimization.

Consider a neural network model with normalization layers (such as batch normalization [3], group normalization [34], layer normalization [35], and others [23]), which introduces scale symmetry in the potential (loss) function $f((1+s)q) = f(q)$. We train this model with gradient descent with a finite learning rate η, momentum β, and weight decay k. We are interested in the evolution of the potential function through the learning dynamics $f(q(t))$. Due to the scale symmetry of the network introduced by the normalization layer, the time evolution on the loss landscape only depends on the directional vector $\hat{q} \equiv q/|q|$ of the parameters throughout training, $f(q(t)) = f(\hat{q}(t))$.

What is the equation of motion for the directional vector? To solve for the dynamics, we first move to a polar coordinate system $q = r\hat{q}$ where $r = |q|$ is the radial norm and $\hat{q}$ is the directional vector. This is akin to how we solve for the motion of a planet in a central field in classical mechanics. From Eq.4, the Lagrangian for this learning system in polar coordinates is,

$$\mathcal{L}_c = e^{\frac{\mu}{m}t} \left[\frac{m}{2}\dot{r}^2 + \left(\frac{m}{2} \left| \dot{\hat{q}} \right|^2 - \frac{k}{2} \right) r^2 - f(\hat{q}) \right] + \lambda(t) \left(|\hat{q}|^2 - 1 \right), \tag{14}$$

where $m = \frac{\eta(1+\beta)}{2}$, $\mu = 1 - \beta$, and $C(\hat{q}) = |\hat{q}|^2 - 1 = 0$ is a Lagrange multiplier enforcing the unit norm constraint. The Lagrangian formulation facilitates coordinate transformations, because the form of the Euler-Lagrange equation is invariant under coordinate transformations. Taking advantage of this fact, we apply the Euler-Lagrange equation to derive the equation of motion for the directional vector as (see SI Sec. C for the derivation)

$$m\ddot{\hat{q}} + \mu\dot{\hat{q}} = -\frac{1}{r^2(t)} \nabla f(\hat{q}), \tag{15}$$

where $\nabla f(\hat{q})$ represents the gradient of f evaluated at $\hat{q}$. Notice the striking similarity of this equation of motion with the equation of motion of the original parameters $m\ddot{q} + \mu\dot{q} = -\nabla f(q)$. This implies that the time evolution of the loss for a network with normalization layers trained by gradient descent is equivalent to the time evolution of the loss for an identical network with unit filter norm trained by an adaptive optimizer which scales the gradients by a factor of $r^2(t)$.

Strikingly, NLD describes the motion of this adaptive scaling factor $r^2(t)$. When $\alpha_t = -\log m$, and $h(x) = \frac{1}{2}|x|^2$, then the charge is $\langle \Delta_h, \partial_s Q \rangle = m\langle \dot{q}, q \rangle = \frac{1}{2} m \partial_t r^2$ and Eq.11, including the effect

of weight decay k, implies

$$m\partial_t^2 r^2 + \mu\partial_t r^2 = -2kr^2 + \frac{2m}{\mu^2 r^2}|\nabla f(\hat{q})|^2, \tag{16}$$

where we kept the first order terms in m and k. In the over-damped regime, ignoring inertia, we get a change in the squared parameter norm $r^2(t)$ as (see SI Sec. D for the derivation):

$$r^2(t) = \sqrt{\frac{2\eta(1+\beta)}{(1-\beta)^3}\int_0^t e^{-\frac{4k}{1-\beta}(t-\tau)}|\nabla f(\hat{q}(\tau))|^2 d\tau + e^{-\frac{4k}{1-\beta}t} r^4(0)}. \tag{17}$$

Each term in this implicit adaptive learning rate has an intuitive meaning. The first integral term keeps track of the recent history of the magnitude of gradient norms $|\nabla f(\hat{q}(\tau))|^2$ that the filters with fixed unit norm receive. The weight decay k controls the time-scale of the short-term memory of the accumulated gradient norms through the exponential kernel $e^{-\frac{4k}{1-\beta}(t-\tau)}$. When $k = 0$ all the gradients will be accumulated resembling an AdaGrad optimizer [36]. Notably, this integral term is proportional to the learning rate η implying that this mechanism of adaptive optimization is only present with a finite learning rate. The second term represents an exponentially decaying memory of initialization. The finite filter norm at initialization $r^4(0)$ prevents instability of learning dynamics avoiding division by zero.

So how does this adaptive scaling of the gradient step sizes compare with hand-designed adaptive optimizers successful in deep learning [36, 37, 38]? Here, we develop a continuous-time model of the explicit adaptive optimizer RMSProp and discover that the functional form of normalization layers' effective learning rate schedule Eq. 17 functionally agrees with that of RMSProp. The RMSProp algorithm is a recursive update rule expressed as $q_{n+1} = q_n - \frac{\eta}{\sqrt{G_n}}\nabla f(q_n)$, where the gradient is scaled by a factor of $\sqrt{G_n}$. The factor G_n keeps track of the history of the gradient norms following $G_{n+1} = \rho G_n + (1-\rho)|\nabla f(q_n)|^2$, where ρ is a hyperparameter. Solving the continuous-time dynamics $\eta\frac{dG}{dt} = -(1-\rho)G(t) + (1-\rho)|\nabla f(q(\tau))|^2$ of the adaptive scaling factor of RMSProp, we obtain

$$\sqrt{G(t)} = \sqrt{\frac{1-\rho}{\eta}\int_0^t e^{-\frac{1-\rho}{\eta}(t-\tau)}|\nabla f(q(\tau))|^2 d\tau + e^{-\frac{1-\rho}{\eta}t} G(0)}. \tag{18}$$

Strikingly, we find that this functional form induced by RMSProp exactly matches with the implicit adaptive learning rate schedule Eq.17 due to normalization layers suggesting potential benefits of the implicit adaptive optimization induced by normalization.

In summary, our theory has yielded the following new insights:

1. The optimization geometry of SGD with a finite step size η, momentum β, and weight decay k together breaks the scale symmetry of the loss (e.g., with normalization layers) and generates an implicit mechanism of adaptive optimization akin to RMSProp.
2. The implicit adaptive optimization mechanism exists when the learning dynamics is in the underdamped regime $m/\mu \propto \eta/(1-\beta) \gg 0$ and thus requires the learning rate to be finite.
3. Momentum significantly amplifies the effect of adaptive optimization by $\frac{1+\beta}{(1-\beta)^3} = 1900$ folds as in Eq. 17 even in the standard setting of $\beta = 0.9$.
4. For scale-invariant parameters, symmetry breaking due to weight decay $1-k$ plays a role similar to the discount factor ρ for the cumulative gradient norm in RMSProp.

Finally, we empirically validate these predictions by training convolutional neural networks with batch normalization (VGG11) on a large data set (Tiny-ImageNet) under various hyperparameter settings as in Fig. 4 (see SI Sec. E). In addition to the standard training with an "unconstrained filter norm" (pink), we perform ablation experiments with a "constrained filter norm" (blue). As a concrete example, consider a convolutional filter $q \in \mathbb{R}^n$ preceding the batch normalization layer. The loss is invariant under the scaling transformation of the filter norm $f((1+s)q) = f(q)$. To test the presence and benefits of implicit adaptive optimization to the learning dynamics of the loss, we freeze the dynamics of the filter norm by repeatedly performing the operation $q(t) \leftarrow \frac{q(t)}{|q(t)|}|q(0)|$ after each step of optimization [33]. This enforces the filter norm to be fixed $|q(t)| = |q(0)|$ throughout training for any t. When trained with a constant small learning rate $\eta \sim 0.001$, the final test accuracy (A-1)

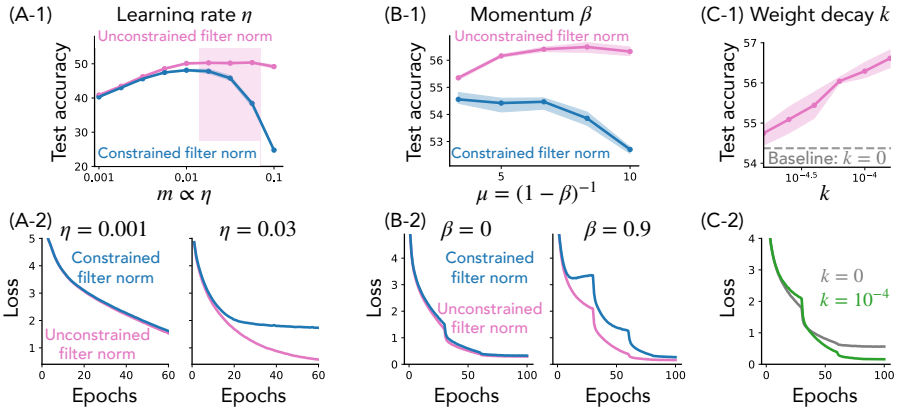

Figure 4: **Broken-symmetry induced dynamics of the filter norm is a key to successful learning.** (VGG-11 models trained on Tiny-ImageNet.) (A-1) The final test accuracy and (A-2) the loss dynamics of models trained at various constant learning rates η with $\beta = 0.9$. (B-1) The final test accuracy and (B-2) the loss dynamics of models trained with various momentum β with learning rate drops. In agreement with our theory, the advantage of implicit adaptive optimization is present when trained with large learning rates and large momentum. (C-1) The final test accuracy and (C-2) the loss dynamics when the model is trained with various weight decay rates k, validating its benefits.

and loss dynamics (A-2) of models with unconstrained filter norm (pink) and constrained filter norm (blue) are almost identical. However, in a large learning rate regime $\eta \sim 0.03$, the models trained with unconstrained filter norm (pink) outperforms the models trained with constrained filter norm (blue) demonstrating the existence and benefits of implicit adaptive optimization. Similarly, we confirm our prediction that the presence of momentum $\beta \sim 0.9$ is essential to amplify the effects of implicit adaptive optimization, as seen in test accuracy (B-1) and loss dynamics (B-2). Finally, we demonstrate the benefits of symmetry breaking due to weight decay k which acts in analogy with the discounting factor ρ for the cumulative gradient norms of RMSProp. Indeed, the final test accuracy (C-1) and the learning dynamics of the loss (C-2) are both enhanced by the presence of weight decay k in agreement with [39]. Overall, the hyperparameters of the modern optimization - learning rate, momentum, and weight decay - each play an important role in breaking the scale symmetry of the loss function and providing an mechanism of implicit adaptive optimization, a key to successful learning in neural networks.

5 Related works and future directions

Finite learning rates, modified gradient flow, and generalization. Recently, many works have demonstrated that an initial large learning rate is a key to successful training [40, 41, 42, 43, 44]. Another line of works [1, 13, 14, 15, 16, 45, 46] have incorporated the effect of a finite learning rate into gradient flow by adding a modification term. A common approach applies modified equation analysis to replace the original loss function f by a modified loss function $\tilde{f} = f + \frac{\eta}{4}|\nabla f|^2$ [1, 14, 15], where the additional term is called *implicit gradient regularization*. Intuitively, this term biases the learning dynamics towards wider minima with smaller gradient norms. To test this intuition empirically, recent works have explicitly added the gradient regularization term $\frac{\eta}{4}|\nabla f|^2$ to the original loss and have shown that the models trained with the modified loss generalize well even with small learning rates [16] or full-batch [17]. In this work, we have taken a complementary approach and applied modified equation analysis *in time* to add an *implicit acceleration* term $\frac{\eta}{2}\ddot{q}$ [1, 13, 45]. When training with a finite learning rate, this implicit acceleration term gives rise to the implicit adaptation of the gradients. Remarkably, the two complementary pictures (i.e., implicit gradient regularization and implicit acceleration) offer a consistent view on how the large learning rates provide mechanisms to properly regularize the norm of gradients to avoid training instabilities. An idea for future work is to directly simulate gradient flow with implicit acceleration, instead of gradient flow with [15, 16, 17] or without [47] implicit gradient regularization, to keep the benefits of finite learning rates while avoiding the computations of the Hessian gradient products.

Geometry of optimization. The geometric property of optimization has a long history of study. For example, natural gradient by Amari [9] and its approximations [24, 48] are invariant under affine transformations including scale transformation due to normalization layers [39]. With the rise of deep learning, there has been recent developments to efficiently approximate and apply natural gradients to large neural networks [24, 26, 48, 49, 50, 51, 52, 53]. When training deep neural networks, we still commonly use gradient-based optimization methods represented by stochastic gradient descent with momentum [12]. An array of adaptive optimization methods [36, 37, 38] have been proposed partially motivated as diagonal approximations of the Fisher Information Matrix (see recent discussions questioning the validity of the approximations [54]). A pioneering work [55] by Neyshabur et al. has identified that gradient descent does not respect the rescale symmetry introduced by ReLU activations, followed by works proposing to solve such discrepancies by enforcing symmetry in the learning dynamics [56, 57]. While above works have always tried to remove such broken symmetries, we demonstrated theoretically and empirically that the broken-symmetry can play an important role in deep learning, thus providing a unique counterexample. A future direction is to extend our analysis to the case of rescale symmetry of the homogemeous activation functions such as ReLU.

Conservation laws in neural networks. Balancing properties of network parameters have been identified in gradient flow dynamics of neural networks with linear [29] and ReLU [30, 31] activation functions. These properties have been leveraged to better understand the role of over-parameterization [29], implicit regularization [30], network pruning [31], continual learning [58], and self-supervised learning [32]. A recent work identified symmetry-induced geometric structures of the loss and studied their roles under gradient flow dynamics with modified loss predicting the parameter dynamics [1]. However, despite the ubiquity of conservation laws in deep learning, no formal connection to the Noether's theorem has been made due to the previous focus on the gradient flow dynamics. Here, we took an alternative path starting from a Lagrangian formulation, which encapsulates gradient flow as the mass-less limit. From this perspective, we unified all the conservation laws under the umbrella of Noether's theorem, and provided its generalization for accelerated and non-Euclidean methods in optimization. In future works, we can harness the generality of our theory to investigate the dynamics of more complex learning rules such as natural gradient descent with or without Nesterov's momentum in combination with any differentiable symmetries.

Auto-rate tuning by normalization layers. Batch normalization [3] and its variants [35, 59] are essential to train *deep* neural network models [60] providing multiple benefits as elegantly reviewed in [61]. An important benefit of normalization layers is to enable stable training with large learning rates [62]. A pioneering work [33] rigorously studied the benefits of monotonically decreasing the effective learning rate in simple setting without momentum nor weight decay. Another line of works noticed the role of weight decay to effectively enforce a higher learning rate [39, 63, 64] inspiring an automatic learning rate scheduler [65]. Li and Arora [66] theoretically studied the setting with weight decay and momentum to design a exponential learning rate schedule that still leads to successful training counter-intuitively. In the present analysis, we leveraged the Lagrangian formulation to handle the complexity of modern practical optimizers and derived an exact solution describing the time-evolution of the effective learning rate in the presence of momentum and weight decay. By developing a continuous-time description of adaptive optimizers in parallel, we discovered that the functional form of the effective learning rate exactly matches with that of RMSProp, where momentum significantly amplifies the effect, and weight decay acts in exact analogy with the discounting factor. In the context of thorough empirical studies on adaptive optimizers [67, 68, 69], our results raise the hypothesis that the benefit of an adaptive optimizers may not be as significant in networks with normalization layers as long as the hyper-parameters, in particular momentum, are properly tuned.

6 Conclusion.

In nature, symmetry governs regularities, while symmetry breaking brings texture [70], examples of which include superconductivity [71], nucleon mass [72], ring attractor neural networks [73, 74, 75], and the Higgs boson [76]. Despite many attempts to bring symmetry to neural networks [2, 55], symmetry breaking has rarely been discussed as a design principle [22], or even treated as an obstacle to successful learning [55, 56, 57]. In this work, we have discovered a novel role of explicit symmetry breaking in learning systems by applying a Lagrangian formulation to neural networks. Overall, understanding not only when symmetries exist, but how they are broken is essential to discover geometric design principles in neural networks.

Acknowledgments and Disclosure of Funding

We thank Max Aalto, Kyosuke Adachi, Yuto Ashida, Fatih Dinc, Surya Ganguli, Kyogo Kawaguchi, Ekdeep Singh Lubana, Akiko Tanaka, Hajime Tanaka, Atsushi Yamamura, and Daniel Yamins for helpful discussions. D.K. thanks the Stanford Data Science Scholars program and NTT Research for support.

References

[1] Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel LK Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=q8qLAbQBupm.

[2] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. URL https://arxiv.org/abs/2104.13478.

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/ioffe15.html.

[4] Andre Wibisono and Ashia C. Wilson. On accelerated methods in optimization, 2015. URL https://arxiv.org/abs/1509.03616.

[5] Lev Davidovich Landau and Evgenii Mikhailovich Lifshitz. *Mechanics: Volume 1*, volume 1. Butterworth-Heinemann, 1976.

[6] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[7] Ashia C Wilson, Benjamin Recht, and Michael I Jordan. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016. URL https://arxiv.org/abs/1611.02635.

[8] E Kanai. On the quantization of the dissipative systems. *Progress of Theoretical Physics*, 3(4): 440–442, 1948. URL https://academic.oup.com/ptp/article/3/4/440/1855867?login=true.

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

[10] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[11] 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*, pages 2510–2518, 2014.

[12] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[13] Nikola B. Kovachki and Andrew M. Stuart. Continuous time analysis of momentum methods. *Journal of Machine Learning Research*, 22(17):1–40, 2021. URL http://jmlr.org/papers/v22/19-466.html.

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

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

[16] Samuel L Smith, Benoit Dherin, David GT Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. *arXiv preprint arXiv:2101.12176*, 2021.

[17] Jonas Geiping, Micah Goldblum, Phillip E Pope, Michael Moeller, and Tom Goldstein. Stochastic training is not necessary for generalization. *arXiv preprint arXiv:2109.14119*, 2021.

[18] Berfin Şimşek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. *arXiv preprint arXiv:2105.12221*, 2021.

[19] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

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

[21] Arthur Jacot, François Ged, Franck Gabriel, Berfin Şimşek, and Clément Hongler. Deep linear networks dynamics: Low-rank biases induced by initialization scale and l2 regularization. *arXiv preprint arXiv:2106.15933*, 2021.

[22] Robert Bamler and Stephan Mandt. Improving optimization for models with continuous symmetry breaking. In *International Conference on Machine Learning*, pages 423–432. PMLR, 2018.

[23] Ekdeep Singh Lubana, Robert P Dick, and Hidenori Tanaka. Beyond batchnorm: Towards a general understanding of normalization in deep learning. *arXiv preprint arXiv:2106.05956*, 2021.

[24] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR, 2015.

[25] Yang Song, Jiaming Song, and Stefano Ermon. Accelerating natural gradient with higher-order invariance. In *International Conference on Machine Learning*, pages 4713–4722. PMLR, 2018.

[26] James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

[27] Kevin Luk and Roger Grosse. A coordinate-free construction of scalable natural gradient. *arXiv preprint arXiv:1808.10340*, 2018.

[28] Emmy Noether. Invariante variationsprobleme, math-phys. *Klasse, pp235-257*, 1918.

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

[30] 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*, pages 384–395, 2018.

[31] Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL `https://proceedings.neurips.cc/paper/2020/file/46a4378f835dc8040c8057beb6a2da52-Paper.pdf`.

[32] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. *arXiv preprint arXiv:2102.06810*, 2021.

[33] Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rkxQ-nA9FX`.

[34] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

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

[36] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[37] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2): 26–31, 2012.

[38] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[39] Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1lz-3Rct7`.

[40] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[41] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv preprint arXiv:1907.04595*, 2019.

[42] Guillaume Leclerc and Aleksander Madry. The two regimes of deep network training. *arXiv preprint arXiv:2002.10376*, 2020.

[43] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

[44] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. *arXiv preprint arXiv:2002.09572*, 2020.

[45] Daniel Kunin, Javier Sagastuy-Brena, Lauren Gillespie, Eshed Margalit, Hidenori Tanaka, Surya Ganguli, and Daniel LK Yamins. Rethinking the limiting dynamics of sgd: modified loss, phase space oscillations, and anomalous diffusion. *arXiv preprint arXiv:2107.09133*, 2021.

[46] Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling sgd with stochastic differential equations (sdes). *arXiv preprint arXiv:2102.12470*, 2021.

[47] Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

[48] Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2016.

[49] Nicolas Le Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. In *NIPS*, pages 849–856. Citeseer, 2007.

[50] James Martens. Deep learning via hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.

[51] Yann Ollivier. Riemannian metrics for neural networks i: feedforward networks. *Information and Inference: A Journal of the IMA*, 4(2):108–153, 2015.

[52] Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

[53] Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. *arXiv preprint arXiv:2002.09018*, 2020.

[54] Frederik Kunstner, Lukas Balles, and Philipp Hennig. Limitations of the empirical fisher approximation for natural gradient descent. *arXiv preprint arXiv:1905.12558*, 2019.

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

[56] Vijay Badrinarayanan, Bamdev Mishra, and Roberto Cipolla. Understanding symmetries in deep networks. *arXiv preprint arXiv:1511.01029*, 2015.

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

[58] Ekdeep Singh Lubana, Puja Trivedi, and Robert P Dick. Rethinking quadratic regularizers: Explicit movement regularization for continual learning. *arXiv preprint arXiv:2102.02805*, 2021.

[59] 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*, pages 901–909, 2016.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[61] Roger Grosse. Lecture 5: Adaptive gradient methods, normalization, and weight decay. *CSC2541 Winter 2021 Topics in Machine Learning: Neural Net Training Dynamics*, 2021. URL `https://www.cs.toronto.edu/~rgrosse/courses/csc2541_2021/`.

[62] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/36072923bfc3cf47745d704feb489480-Paper.pdf`.

[63] 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*, pages 2160–2170, 2018.

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

[65] Aitor Lewkowycz. How to decay your learning rate. *arXiv preprint arXiv:2103.12682*, 2021.

[66] Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rJg8TeSFDH`.

[67] Dami Choi, Christopher J Shallue, Zachary Nado, Jaehoon Lee, Chris J Maddison, and George E Dahl. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2019.

[68] Robin M Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley-benchmarking deep learning optimizers. In *International Conference on Machine Learning*, pages 9367–9376. PMLR, 2021.

[69] Zachary Nado, Justin M Gilmer, Christopher J Shallue, Rohan Anil, and George E Dahl. A large batch optimizer reality check: Traditional, generic optimizers suffice across batch sizes. *arXiv preprint arXiv:2102.06356*, 2021.

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

[71] H Suhl, BT Matthias, and LR Walker. Bardeen-cooper-schrieffer theory of superconductivity in the case of overlapping bands. *Physical Review Letters*, 3(12):552, 1959.

[72] Yoichiro Nambu and Giovanni Jona-Lasinio. Dynamical model of elementary particles based on an analogy with superconductivity. i. *Physical review*, 122(1):345, 1961.

[73] Shun-ichi Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological cybernetics*, 27(2):77–87, 1977.

[74] Rani Ben-Yishai, R Lev Bar-Or, and Haim Sompolinsky. Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences*, 92(9):3844–3848, 1995.

[75] Sung Soo Kim, Hervé Rouault, Shaul Druckmann, and Vivek Jayaraman. Ring attractor dynamics in the drosophila central brain. *Science*, 356(6340):849–853, 2017.

[76] Peter Ware Higgs. Broken symmetries, massless particles and gauge fields. *Phys. Lett.*, 12: 132–133, 1964.

[77] 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.

[78] 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.

Supplementary Information

A The principle of least action and the Euler-Lagrange equation

Here, we review the principle of least action and the derivation of the Euler-Lagrange equation [5]. The principle of least (stationary) action states that the time evolution of the position of a particle $q(t) \in \mathbb{R}^N$ with boundary conditions ($q(t_0) = a$ and $q(t_1) = b$) is the function that minimizes the action,

$$S[q] = \int_{t_0}^{t_1} \mathcal{L}(q, \dot{q}, t) dt.$$

Now, let us derive the differential equation that gives a solution to the variational problem. Assuming q is the solution, then it implies that any small variation $q \to q + \delta q$ will increase the value of the action. Thus, the stationary condition for the solution is,

$$\delta S = \int_{t_1}^{t_2} \mathcal{L}(q + \delta q, \dot{q} + \delta \dot{q}, t) dt - \int_{t_1}^{t_2} \mathcal{L}(q, \dot{q}, t) dt = \int_{t_1}^{t_2} \left(\frac{\partial \mathcal{L}}{\partial q} \delta q + \frac{\partial \mathcal{L}}{\partial \dot{q}} \delta \dot{q} \right) dt = 0. \tag{19}$$

By integrating the second term by parts, we get

$$\delta S = \left[\frac{\partial \mathcal{L}}{\partial \dot{q}} \delta q \right]_{t_1}^{t_2} + \int_{t_1}^{t_2} \left(\frac{\partial \mathcal{L}}{\partial q} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} \right) \delta q dt = 0. \tag{20}$$

The first term is zero due to the fixed boundary conditions, $\delta q(t_1) = \delta q(t_2) = 0$. Thus, the second term must vanish for arbitrary variation δq. This condition yields the Euler-Lagrange equation,

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} = \frac{\partial \mathcal{L}}{\partial q}. \tag{21}$$

B The derivation of the Noether's learning dynamics

Noether's learning dynamics. *Consider the learning dynamics of a parameter vector $q \in \mathbb{R}^N$ governed by the Bregman Lagrangian $\mathcal{L}(q, \dot{q}, t) = e^{\gamma_t} \left(e^{\alpha_t} D_h(q + e^{-\alpha_t} \dot{q}, q) - e^{\alpha_t + \beta_t} f(q) \right)$. If the loss function $f(q)$ is invariant under a one-parameter family of differentiable maps $q(t) \to Q(q(t), s)$, i.e., $f(Q(q(t), s)) = f(q)$ for all (q, s), then the learning dynamics obeys the following equality,*

$$\frac{d}{dt} \left\langle \Delta_h, \frac{\partial Q}{\partial s} \right\rangle + \dot{\gamma}_t \left\langle \Delta_h, \frac{\partial Q}{\partial s} \right\rangle = \left\langle \Delta_h, \frac{\partial \dot{Q}}{\partial s} \right\rangle + e^{\alpha_t} \left\langle \Delta_h - e^{-\alpha_t} \nabla^2 h(q) \dot{q}, \frac{\partial Q}{\partial s} \right\rangle, \tag{22}$$

where Δ_h is the Bregman momentum defined as $\Delta_h \equiv e^{-\gamma_t} \partial_{\dot{q}} \mathcal{L} = \nabla h(q + e^{-\alpha_t} \dot{q}) - \nabla h(q)$.

Derivation. Here, we derive the Noether's learning dynamics by applying Noether's theorem to the Bregman Lagrangian. A general form of the Noether's theorem relates the dynamics of Noether charge $\frac{d}{dt} \left\langle \frac{\partial \mathcal{L}}{\partial \dot{q}}, \frac{\partial Q}{\partial s} \right\rangle$ to the change of Lagrangian at infinitesimal symmetry transformation $\frac{\partial \mathcal{L}}{\partial s}|_{s=0}$ evaluated at the identity transformation as,

$$\frac{\partial \mathcal{L}}{\partial s} = \left\langle \frac{\partial \mathcal{L}}{\partial q}, \frac{\partial Q}{\partial s} \right\rangle + \left\langle \frac{\partial \mathcal{L}}{\partial \dot{q}}, \frac{\partial \dot{Q}}{\partial s} \right\rangle = \frac{d}{dt} \left\langle \frac{\partial \mathcal{L}}{\partial \dot{q}}, \frac{\partial Q}{\partial s} \right\rangle,$$

where we plugged in the Euler-Lagrange equation $\frac{\partial \mathcal{L}}{\partial q} = \frac{d}{dt} \left\langle \frac{\partial \mathcal{L}}{\partial \dot{q}} \right\rangle$ and performed integration by parts. Since the loss function $f(q)$ is invariant under the differentiable transformation $q(t) \to Q(q(t), s)$, we have $\frac{\partial}{\partial s} f(Q) = 0$. Thus, we evaluate the dynamics of the Noether charge driven by the kinetic symmetry breaking,

$$\frac{d}{dt} \left\langle \frac{\partial \mathcal{L}}{\partial \dot{q}}, \frac{\partial Q}{\partial s} \right\rangle = e^{\gamma_t} \frac{\partial T_h}{\partial s}, \tag{23}$$

where T_h is the kinetic energy of learning defined as,

$$T_h(q, \dot{q}, t) = e^{\alpha_t} \left(h(q + e^{-\alpha_t}\dot{q}) - h(q) - \langle \nabla h(q), e^{-\alpha_t}\dot{q} \rangle \right).$$

For immediate use, we first evaluate the derivative of the kinetic energy T_h with respect to q and $\dot{q}$ as,

$$\frac{\partial T_h}{\partial q} = e^{\alpha_t} \left(\nabla h(q + e^{-\alpha_t}\dot{q}) - \nabla h(q) - \langle \nabla^2 h(q), e^{-\alpha_t}\dot{q} \rangle \right) = e^{\alpha_t} \left(\Delta_h - \langle \nabla^2 h(q), e^{-\alpha_t}\dot{q} \rangle \right),$$

and

$$\frac{\partial T_h}{\partial \dot{q}} = \nabla h(q + e^{-\alpha_t}\dot{q}) - \nabla h(q) = \Delta_h. \tag{24}$$

By evaluating the left hand side of Eq. 23, we get

$$\frac{d}{dt}\left\langle \frac{\partial \mathcal{L}}{\partial \dot{q}}, \frac{\partial Q}{\partial s} \right\rangle = \frac{d}{dt}\left(e^{\gamma_t} \left\langle \frac{\partial T_h}{\partial \dot{q}}, \frac{\partial Q}{\partial s} \right\rangle \right) = \frac{d}{dt}\left(e^{\gamma_t} \left\langle \Delta_h, \frac{\partial Q}{\partial s} \right\rangle \right) = e^{\gamma_t}\left(\dot{\gamma}_t \left\langle \Delta_h, \frac{\partial Q}{\partial s} \right\rangle + \frac{d}{dt}\left\langle \Delta_h, \frac{\partial Q}{\partial s} \right\rangle \right).$$

By evaluating the right hand side of Eq. 23, we get

$$e^{\gamma_t}\frac{\partial T_h}{\partial s} = e^{\gamma_t}\left(\left\langle \frac{\partial T_h}{\partial q}, \frac{\partial Q}{\partial s} \right\rangle + \left\langle \frac{\partial T_h}{\partial \dot{q}}, \frac{\partial \dot{Q}}{\partial s} \right\rangle \right) = e^{\gamma_t}\left[e^{\alpha_t}\left\langle \Delta_h - \nabla^2 h(q) e^{-\alpha_t}\dot{q}, \frac{\partial Q}{\partial s} \right\rangle + \left\langle \Delta_h, \frac{\partial \dot{Q}}{\partial s} \right\rangle \right].$$

Finally, by equating the left and right sides, and then dividing both sides by e^{γ_t}, we obtain the Noether's learning dynamics. □

C Motion in scale-symmetric potential

In this section, we consider motion of a particle in scale-symmetric potential governed by the Lagrangian,

$$\mathcal{L}(q, \dot{q}, t) = e^{\frac{\mu}{m}t}\left(\frac{m}{2}|\dot{q}|^2 - \left(f(q) + \frac{k}{2}|q|^2 \right) \right), \tag{25}$$

where $q \in \mathbb{R}^N$ is a vector representing the general coordinate, m is the mass, μ is the damping coefficient, $f(q)$ is the potential function with scale symmetry, and k is the spring constant for the quadratic potential function. With the parameterization of $m = \frac{\eta}{2}(1+\beta)$ and $\mu = 1 - \beta$, this dynamics is equivalent to the learning dynamics of the scale-invariant parameter vector q under gradient descent with a finite learning rate η, momentum β, and weight decay k.

C.1 Equations of motion

First, we leverage the scale symmetry of the potential (loss) function by separating the coordinate as $q = r\hat{q}$, where r is the radial length of the parameter vector q, and $\hat{q}$ is the unit vector representing the direction. The time derivative is $\dot{q} = \dot{r}\hat{q} + r\dot{\hat{q}}$, and the Lagrangian can be re-written with this new coordinate as,

$$\begin{aligned} \mathcal{L} &= e^{\frac{\mu}{m}t}\left(\frac{m}{2}\left(\dot{r}^2 + 2r\dot{r}\overbrace{\left\langle \hat{q}, \dot{\hat{q}} \right\rangle}^{=0} + r^2\left|\dot{\hat{q}}\right|^2 \right) - \left(f(\hat{q}) + \frac{k}{2}r^2 \right) \right) + \lambda(t)(|\hat{q}|^2 - 1) \\ &= e^{\frac{\mu}{m}t}\left(\frac{m}{2}\dot{r}^2 + \left(\frac{m}{2}\left|\dot{\hat{q}}\right|^2 - \frac{k}{2} \right) r^2 - f(\hat{q}) \right) + \lambda(t)(|\hat{q}|^2 - 1). \end{aligned} \tag{26}$$

Now, we harness the covariant property of the Lagrangian formulation, i.e., it preserves the form of the Euler-Lagrange equations after change of coordinates, to obtain the radial and angular Euler-Lagrange equations,

Radial Euler-Lagrange equation: $\partial_r \mathcal{L} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{r}}\right)$,

$$m\ddot{r} + \mu\dot{r} = \left(m\left|\dot{\hat{q}}\right|^2 - k \right) r. \tag{27}$$

Derivation. We evaluate $\partial_r \mathcal{L} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{r}}\right)$ by substituting the Lagrangian (Eq. 26). By evaluating the left hand side, we get

$$\frac{\partial \mathcal{L}}{\partial r} = e^{\frac{\mu}{m}t}\left(m\left|\dot{\hat{q}}\right|^2 - k\right) r.$$

By evaluating the right hand side, we get

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{r}}\right) = \frac{d}{dt}\left(me^{\frac{\mu}{m}t}\dot{r}\right) = \mu e^{\frac{\mu}{m}t}\dot{r} + me^{\frac{\mu}{m}t}\ddot{r}.$$

By equating the left and right hand sides and dividing both sides by $e^{\frac{\mu}{m}t}$, we obtain the radial Euler-Lagrange equation.

□

Angular Euler-Lagrange equation for $\hat{q}$: $\partial_{\hat{q}} \mathcal{L} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\hat{q}}}\right)$

$$m\ddot{\hat{q}} + \mu\dot{\hat{q}} = -\frac{1}{r^2}\nabla f(\hat{q}). \tag{28}$$

Derivation. We evaluate $\partial_{\hat{q}} \mathcal{L} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\hat{q}}}\right)$ by substituting the Lagrangian (Eq. 26). By evaluating the left hand side, we get

$$\frac{\partial \mathcal{L}}{\partial \hat{q}} = -e^{\frac{\mu}{m}t}\nabla f(\hat{q}) + 2\lambda(t)\hat{q}.$$

By evaluating the right hand side, we get

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\hat{q}}}\right) = \frac{d}{dt}\left(e^{\frac{\mu}{m}t}m\dot{\hat{q}}r^2\right) = e^{\frac{\mu}{m}t}\left(\mu r^2\dot{\hat{q}} + mr^2\ddot{\hat{q}} + 2mr\dot{r}\dot{\hat{q}}\right).$$

By equating the both sides and rearranging terms, we get,

$$\dot{\hat{q}} = \left(1 + \frac{2m\dot{r}}{\mu r}\right)^{-1}\left(-\frac{1}{\mu r^2}\nabla f(\hat{q}) - \frac{m}{\mu}\ddot{\hat{q}} + \frac{2\lambda(t)}{\mu r^2}e^{-\frac{\mu}{m}t}\hat{q}\right).$$

Now, consider the limits of $m \to 0$ and $k \to 0$, which are justified in practice with the small learning rate $m \propto \eta$ and weight decay k. In this limit, the $\dot{r}$ becomes small

$$\dot{r} = \frac{1}{\mu}\left(m|\dot{\hat{q}}|^2 r - kr - m\ddot{r}\right) \sim O(m) + O(k),$$

and $m\dot{r} \sim O(m^2) + O(mk)$ will not be the leading order terms. Thus, by only keeping the first order terms in m and k, we get

$$m\ddot{\hat{q}} + \mu\dot{\hat{q}} = -\frac{1}{r^2}\nabla f(\hat{q}).$$

□

C.2 Limiting dynamics

The derived Euler-Lagrange equations accurately describe deep learning dynamics in realistic settings with a finite learning rate η, momentum β, and weight decay k. Here, we demonstrate the power and accuracy of the Euler-Lagrange equations by deriving important results in the literature that have already been verified empirically. Unlike existing works using a discrete-time framework, our continuous-time framework eliminates all the tedious algebraic calculations and allows us to obtain the results immediately.

Angular velocity. First, by assuming a steady state of radial dynamics $\dot{r} = 0$ in equation Eq.27, we immediately obtain a condition for angular velocity $m|\dot{\hat{q}}|^2 - k = 0$, which then yields a relationship

$$|\dot{\hat{q}}|^* = \sqrt{\frac{k}{m}} = \sqrt{\frac{2k}{\eta(1+\beta)}}. \tag{29}$$

By discretizing this continuous-time expression $|\hat{q}(t+\eta) - \hat{q}(t)| = \sqrt{\frac{2\eta k}{1+\beta}}$ we get a relationship that is consistent with the literature [77].

Radius. Similarly, by substituting a relationship from Eq.28, $\dot{\hat{q}} = -\frac{1}{\mu r^2}\nabla f(\hat{q}) - \frac{m}{\mu}\ddot{\hat{q}}$, into Eq.27 with the steady norm assumption, we obtain a condition for a steady radius r^*,

$$r^* = \sqrt[4]{\frac{m}{k\mu^2}}\sqrt{|\nabla f(\hat{q})|} = \sqrt[4]{\frac{\eta(1+\beta)}{2k(1-\beta)^2}}\sqrt{|\nabla f(\hat{q})|}, \tag{30}$$

where we kept only the leading order in m.

Learning dynamics. Plugging this expression obtained from the steady-state condition of Eq.27 back into Eq.28, we find those steady-state learning dynamics is driven by the normalized gradients scaled by intrinsic learning rate as,

$$m\ddot{\hat{q}} + \mu\dot{\hat{q}} = -\sqrt{\frac{k\mu^2}{m}}\frac{\nabla f(\hat{q})}{|\nabla f(\hat{q})|}, \tag{31}$$

consistent with results derived in [78] in the discrete-time setting.

D Implicit adaptive optimization by normalization layers

Here, we ignore the inertia term in Eq. 16, assuming that the mass (learning rate) is finite but small and the learning dynamics is in the overdamped regime. In this setting, the Eq. 16 reduces to,

$$\frac{d}{dt}r^2 = -\frac{2k}{\mu}r^2 + \frac{2m}{\mu^3 r^2}|\nabla f(\hat{q})|^2. \tag{32}$$

In the setting of gradient descent with no momentum ($\mu = 1-\beta = 1$ and $m = \eta/2$), this expression simplifies to the one previously obtained with the *modified loss* approach in Eq. 21 of [1] and Eq. 35 of [46].

To solve this differential equation, first we multiply both sides by $2r^2$ to get,

$$2r^2\frac{d}{dt}r^2 = -\frac{4k}{\mu}(r^2)^2 + \frac{4m}{\mu^3}|\nabla f(\hat{q})|^2.$$

Rearranging terms and multiplying both sides by $e^{\frac{4k}{\mu}t}$ gives,

$$e^{\frac{4k}{\mu}t}\frac{d}{dt}(r^2)^2 + \frac{4k}{\mu}e^{\frac{4k}{\mu}t}(r^2)^2 = \frac{4m}{\mu^3}e^{\frac{4k}{\mu}t}|\nabla f(\hat{q})|^2.$$

By integrating both sides,

$$\left[(r^2)^2 e^{\frac{4k}{\mu}t}\right]_0^t = \frac{4m}{\mu^3}\int_0^t e^{\frac{4k}{\mu}\tau}|\nabla f(\hat{q}(\tau))|^2 d\tau,$$

and rearranging terms, we get

$$r^2(t) = \sqrt{\frac{4m}{\mu^3}\int_0^t e^{-\frac{4k}{\mu}(t-\tau)}|\nabla f(\hat{q}(\tau))|^2 d\tau + e^{-\frac{4k}{\mu}t}r^4(0)}. \tag{33}$$

E Experimental details

All the experiments were run using the PyTorch code base.

Dataset. We used Tiny ImageNet dataset to generate all the empirical figures in this work. The Tiny ImageNet dataset consists of 100,000 training images at a resolution of 64×64 spanning 200 classes.

Model. We use standard VGG-11 models for all our experiments with a batch normalization layer between every convolutional layer and its activation function.

Hyperparameters. The key hyperparameters we used are listed with each figure. Unless specified otherwise, the standard hyperparameters we used are the momentum 0.9, train batch size 128, weight decay 10^{-4}, where we drop the initial learning rate by a factor of ten after 30, 60, and 80 epochs.