A Space-Time Approach to Analyzing Stochastic Gradient Descent

Anonymous Authors[1]

Abstract

We harness Feynman Diagrams to reason about Stochastic Gradient Descent (SGD) at small learning rates η. We derive dynamical laws allowing us to: construct a regularizer causing large-batch GD to emulate small-batch SGD; exhibit a non-conservative entropic force driving SGD; generalize the Akaike Info. Criterion (AIC) to a smooth quantity liable to descent; and quantify how SGD differs from the popular approximation SDE. We verify our predictions on artificial data and convnets for CIFAR-10 and Fashion-MNIST.

1. Introduction

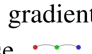
This paper studies the dynamics of SGD. Mainly, how does gradient noise bias learning? Practitioners benefit from the intuition that SGD approximates noiseless GD. We present a novel framework for refining this intuition to account for noise. For instance, we show that noise systematically pushes SGD toward flatter minima. We prove our predictions for small learning rate and our experiments show that even a single evaluation of our force laws suffices to predict SGD's motion through macroscopic timescales, e.g. long enough to decrease error by 0.5 percentage points.

Departing from prior work, we model discrete time and hence non-Gaussian noise. Indeed, we give the finite-time, finite-learning-rate corrections to continuous-time approximations such as ordinary and stochastic differential equations (ODE, SDE). We thus quantify how epoch number and batch size affect test loss. Our theory of noise recommends two novel regularizers that respectively induce GD to mimic SGD and help to tune hyperparameters such as l_2 coefficients. We verify our predictions on landscapes including CIFAR-10 convnets (details in Appendix **??**).

Our underlying formalism is especially novel: we interpret SGD as a superposition of several concurrent interactions between weights and data, each represented by a diagram that echoes the visual schemata of Feynman (1949); Penrose (1971). This viewpoint offers not only quantitative predictions but also qualitative insight, e.g. that, for small learning rates, inter-epoch shuffling does not affect expected test loss. We believe that our diagram method is an elegant and general tool for studying stochastic optimization, especially on short timescales. Our conclusion discusses Hessian methods and natural GD as low-hanging fruit for future work.

1.1. Overview of the perturbative approach

Consider running SGD on N training points for T steps with learning rate η, starting at a weight θ_0. Our method expresses the expectation (over randomly sampled train sets) of quantities such as the final weight (or test or train loss) as a sum of diagrams, where each diagram evaluates to a statistic of the loss landscape at initialization. Diagrams with e edges contribute only $O(\eta^e)$ to the quantities of interest, so for small η we sum only the few-edged diagrams and incur an $o(\eta^e)$ error term.

The rule for evaluating diagrams is that each degree-d node evaluates to the dth derivative of the test loss l at θ_0. The edges indicate the order in which those derivatives are multiplied. Most simply, $= l(\theta_0)$, a 0th derivative. The diagram * evaluates to the dot product ηGG, where $G = \nabla l$ is the gradient of the expected loss l, evaluated at θ_0. Likewise,  $= \eta^2 GHG$, where $H = \nabla\nabla l$ is the hessian. And $= \eta^3 GGJG$, where $J = \nabla\nabla\nabla l$ is l's third derivative.

A diagram tells us about the loss landscape but not about SGD parameters such as T or inter-epoch shuffling. We summarize those parameters as sets of pairs (n, t), one for each participation of the nth datapoint in the tth update. Full-batch GD will have NT many pairs, for instance, while singeleton-batch SGD will have T many pairs.

Each of a diagram's nodes abstractly represents an event at such a pair, and we may "concretize" a diagram by assigning to each node a specific pair (n, t). We will intuitively interpret a concretized edge from (n, t) to (n', t') (from left to right) as depicting information flow from train point n at

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

*We color nodes for convenient reference (e.g. to a diagram's "green nodes"). As mere labels, colors lack mathematical meaning.

time t to train point n' at time t'. Thus, we permit only concretizations whose edges have $t < t'$. The rightmost node represents measurement at test time, so we do not assign a pair (n, t) to it; it does not participate in $t < t'$ constraints.

Theorem (Theorem 1, Informal)**.** *Fix SGD parameters, namely N, T, a batch size B, and a deterministic routine to sample each batch from a train set. Sum the diagrams with at most d edges, where a diagram with e edges and c many concretizations is weighted by $c/(-B)^e$. This sum agrees with SGD's expected final test loss to order $o(\eta^d)$.*

Example 1. What is SGD's expected test loss to order η^1? There are two diagrams with ≤ 1 edges: $= l(\theta_0)$ and $= \eta GG$. For SGD with batchsize 1, has T concretizations, since its rightmost node must represent the test measurement and its other node can represent any of T many (n, t) pairs. By the Theorem, the answer is $l(\theta_0) - T \cdot \eta GG + o(\eta^1)$.

Example 1 is well-known (e.g. Nesterov (2004)). Indeed, it quantifies the intuition that, in each of T steps, SGD moves the weight by ηG and hence decreases the loss by ηGG. The expression is exact for a noiseless linear landscape, but, because it fails to model how gradients depend on the current weight (curvature) or on the current train point (noise), it is typically an approximation. Our diagrams beyond correct the expression by modeling curvature and noise.

Like Example 1, our predictions depend only on loss data near θ_0 and hence break down after the weight moves far from initialization. Our theory thus best applies to small-movement contexts, whether for long times near a minimum or for short times in general.* For instance, we analyze SGD overfitting near a minimum (Corollary 2). Invoking Theorem 2 to find T so large large that SGD senses curvature and noise, yet small enough for our theory to hold, we analyze how curvature and noise — and not just gradients — repel or attract the evolving weight (Corollary 1).

1.2. Concretizations as embeddings into spacetime

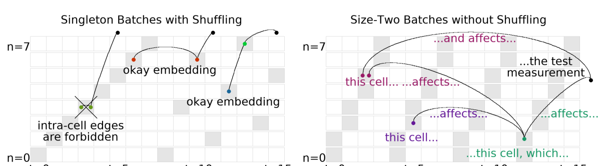


Figure 1. **Diagrams in Spacetime Depict SGD's Subprocesses.** Two spacetimes with $N = 8, T = 16$. **Left**: Batchsize $B = 1$ with inter-epoch shuffling. Embeddings, legal and illegal, of , , and . **Right**: Batchsize $B = 2$ without inter-epoch shuffling. Interpretation of an order η^4 diagram embedding.

* It is routine to check that SGD dynamics can efficiently simulate a Turing Machine and thus that general long-time prediction should be intractable.

To visualize concretizations, we draw (n, t) pairs as shaded cells in an $N \times T$ grid. A concretization is then an embedding of nodes to shaded cells. The $t < t'$ constraint then forbids intra-cell edges (Figure 1 left), and we may interpret each edge as an effect of the past on the future (right). We call an SGD run's set of shaded cells its "spacetime".

1.3. A first look at curvature and noise

Intuitively, our diagrams' edges depict higher derivatives and hence the test loss l's curvature. However, to study noise and generalization, we need to represent how different datapoints n induce different loss functions l_n. The test loss l is then the expectated value of l_n. We depict correlations and hence noise by a new structure: fuzzy "ties". For example, and are two valid and distinct diagrams. Fuzzy ties determine which derivatives occur within the same expectation, so we have

$$\triangleq \eta^2 \mathbb{E}[\nabla l_n] \, \mathbb{E}[\nabla\nabla l_n] \, \mathbb{E}[\nabla l_n] = \eta^2 GHG = \eta^2 \frac{\nabla (GG)}{2} G$$

and, writing C for the covariance of gradients,

$$\triangleq \eta^2 \mathbb{E}[\nabla l_n \nabla\nabla l_n] \, \mathbb{E}[\nabla l_n] = \eta^2 \frac{\nabla (GG + C)}{2} G$$

The rule is that nodes in the same fuzzy-tie connected component occur in the same expectation brackets. Since fuzzy ties depict correlations, we demand that each concretization of a diagram sends any two fuzzily-tied nodes to pairs $(n, t), (n, t')$ that share a train point index n.

Example 2. When $N = T$, then singleton-batch SGD permits permits no concretizations of , since the edge constraint $t < t'$ conflicts with the tie constraint $n = n'$ when, as in this case, the permitted (n, t) pairs comprise a bijection between ns and ts.

Example 3. By constrast, when $N = T$, full-batch GD permits $N\binom{N}{2}$ many concretizations of , since all NT possible (n, t) pairs occur. Those concretizations $(n, t), (n, t')$ have as close analogues the concretizations $(n, t), (n', t')$ in Example 2's setting of the tie-less diagram .

Comparing the two examples above reveals a difference between batchsize-1 and batchsize-N descent for $N = T$: by the Main Theorem, the latter incurs an additional test loss

$$\frac{c}{(-B)^e} \left(\quad - \quad \right) = \overset{algebra}{\cdots} = \frac{\eta^2 (N-1)}{4} G \nabla C$$

It turns out that is the only 2-edged diagram whose concretizations in SGD and GD differ. Thus, this test loss difference between SGD and GD is correct to order η^2.

The above generalizes Roberts (2018)'s $T = 2$ result, proved without diagrams, to arbitrary T. In principle, one could avoid diagrams completely by direct use of our Key Lemma (stated in the Appendix). However, as demonstrated in the Appendix, counting concretizations of diagrams streamlines calculation, yielding arguments 4 times shorter and arguably more insightful conceptual than direct perturbation. The more complicated the computation, the greater the savings of using diagrams.

2. Background and Notation

2.1. Loss landscape

We henceforth fix a loss landscape on a weight space $\mathcal{M}$, i.e. a distribution over smooth functions $l_n : \mathcal{M} \to \mathbb{R}$ whose mean we call l. We refer both to n and to l_n as *datapoints*. We assume the regularity conditions listed in Appendix **??**, for instance that l, l_n are analytic and that all moments exist.

For example, these conditions admit tanh networks with cross entropy loss on bounded data — and with arbitrary weight sharing, skip connections, soft attention, dropout, batch-normalization with disjoint batches, and weight decay.

2.2. Tensor conventions

We use $G_\mu, H_{\mu\nu}, J_{\mu\nu\lambda}$ for the first, second, and third derivatives of l and $C_{\mu\nu}$ for the covariance of gradients. By convention, repeated Greek indices are implicitly summed: if A_μ, B^μ are the coefficients of a covector A and a vector B[*], indexed by basis elements μ, then $A_\mu B^\mu \triangleq \sum_\mu A_\mu \cdot B^\mu$. To expedite dimensional analysis, we regard the learning rate as an inverse metric $\eta^{\mu\nu}$ that converts a gradient covector into a vector displacement (Bonnabel, 2013). We use η to *raise* indices. In $H^\mu_\lambda \triangleq \eta^{\mu\nu} H_{\nu\lambda}$, for instance, η raises one of $H_{\mu\nu}$'s indices. Another example is $C^\mu_\mu \triangleq \sum_{\mu\nu} \eta^{\mu\nu} \cdot C_{\nu\mu}$. Standard syntactic constraints make manifest which expressions transform naturally with respect to optimization dynamics.

We say two expressions *agree to order* η^d when their difference, divided by some homogeneous degree-d polynomial of η, tends to 0 as η shrinks. Their difference is then $\in o(\eta^d)$.

2.3. SGD terminology

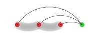
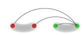
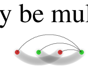
SGD decreases an unknown objective l via T steps of discrete-time η-steepest[†] descent on noisy estimates of l.

We describe SGD in terms of N, T, B, E, M: N counts training points, T counts updates, B counts points per batch, $E = TN/B$ counts epochs, and $M = E/B = T/N$.[‡] SGD then learns from a train set $(l_n : 0 \le n < N)$ via $T = NM$ updates of the form:

$$\theta^\mu \leftsquigarrow \theta^\mu - \eta^{\mu\nu} \nabla_\nu \left(\frac{1}{B} \sum_{n \in \mathcal{B}} l_n(\theta) \right)$$

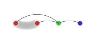
We write l_t for the loss $\frac{1}{B} \sum_{\mathcal{B}} \cdots$ over the tth batch. The cases $B = 1$ and $B = N$ we call *pure SGD* and *pure GD*. The $M = 1$ case of pure SGD we call *vanilla SGD*.

2.4. Diagrams and embeddings

Definition 1 (Diagrams)**.** A diagram is a finite rooted tree equipped with a partition of nodes. We draw the tree using thin "edges". By convention, we draw each node to the right of its children; the root is thus always rightmost. We draw the partition by connecting the nodes within each part via fuzzy "ties". For example,  has 2 parts. We insist on using as few fuzzy ties as possible so that, if d counts edges and c counts ties, then $d + 1 - c$ counts parts. There may be multiple ways to draw a single diagram, e.g.  = .

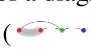
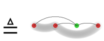
Definition 2 (Evaluating a Diagram)**.** In the context of a loss landscape and an initial weight θ_0 a diagram evaluates to the expectation (over all i.i.d. of datapoints to parts) of a product of derivatives, one dth derivative $\nabla^d l(\theta_0)$ for each degree-d node. Each edge denotes a contraction of its two nodes by the inverse metric η. For example,

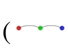
 $\triangleq \mathbb{E}_{n,n'} \left[(\nabla_\mu l_n)(\nabla^\mu l_{n'}) \right](\theta_0)$

 $\triangleq \mathbb{E}_{n,n',n''} \left[(\nabla_\mu l_n)(\nabla_\nu l_n)(\nabla^\mu \nabla^\nu \nabla_\lambda l_{n'})(\nabla^\lambda l_{n''}) \right](\theta_0)$

We write value(D) for a diagram D's value, or D when clear.

Definition 3 (Embedding a Diagram into Spacetime)**.** An embedding of a diagram into a spacetime is an assignment of that diagram's non-root nodes to pairs (n, t) such that each node occurs at a time t' strictly after each of its children and such that two nodes occupy the same row n if and only if they inhabit the same part of D's partition.

Definition 4 (Fuzzy Outlines Denote Noise's Net Effect)**.** We may join any two parts $p, \tilde{p}$ of a diagram D to obtain a new diagram $D_{p\tilde{p}}$. For instance, ()$_{\text{red blue}}$ $\triangleq$ . Since fuzzy ties denote correlation and noise, differences such as $D_{p\tilde{p}} - D$ quantify noise's net effect. So, for convenience, we define a diagram with fuzzy *outlines* as the difference between its fuzzy tied and untied versions, e.g.:

 $\triangleq$ ()$_{\text{green blue}}$ $-$ $=$ $-$

3. Diagram Calculus for SGD

3.1. Recipe for SGD's test loss and generalization

Our main tool is proved in Appendix **??**:

[*] Vectors/covectors are also called column/row vectors.

[†] To define "steepness" requires a metric on l's domain. We will consider all metrics by Taylor expanding an (inverse) metric $\eta^{\mu\nu}$ around 0.

[‡] Since η, N, M determine SGD's final loss on a noiseless, linear landscape, it is natural to compare SGD variants of equal M.

Theorem 1 (Test Loss as a Path Integral)**.** *For all T: for η sufficiently small, SGD's expected test loss is*

$$\sum_{D} \sum_{\textit{embeddings } f} \frac{1}{\left|\mathrm{Aut}_f(D)\right|} \frac{\mathrm{value}(D)}{(-B)^{|\mathrm{edges}(D)|}}$$

Here, D is a diagram whose root r does not participate in any fuzzy edge, f is an embedding of D into spacetime, and $\left|\mathrm{Aut}_f(D)\right|$ counts the graph-automorphisms of D that preserve f's assignment of nodes to cells. If we replace D by $\left(-\sum_{p\in \mathrm{parts}(D)}(D_{rp} - D)/N\right)$, where r is D's root, we obtain the expected generalization gap (test minus train loss).

Proposition 1 (Specialization to Vanilla SGD)**.** *The order η^d contribution to the expected test loss of one-epoch SGD with singleton batches is:*

$$\frac{(-1)^d}{d!} \sum_{D} |\mathrm{ords}(D)| \binom{N}{P-1} \binom{d}{d_0, \cdots, d_{P-1}} \mathrm{value}(D)$$

where D ranges over d-edged diagrams whose root does not participate in any fuzzy edge and each of whose parts contains none of its nodes' ancestors. Here, D's parts have sizes $d_p : 0 \leq p \leq P$, and $|\mathrm{ords}(D)|$ counts the total orderings of D s.t. children precede parents and parts are contiguous. Theorem 1's modification for the gen. gap still holds.

By Proposition 1, a diagram with d thin edges and f fuzzy ties (hence $d + 1 - c$ parts), contributes $\Theta\left((\eta T)^d T^{-c}\right)$ to vanilla SGD's test loss.

Intuitively, ηT measures the physical time of descent and T^{-1} measures the coarseness of time discretization. We thus obtain a double series in $(\eta T)^d T^{-c}$; the $c = 0$ terms correspond to a noiseless, discretization-agnostic (hence ODE) approximation to SGD, the the remaining terms model time-discretization and noise. See Table 1.

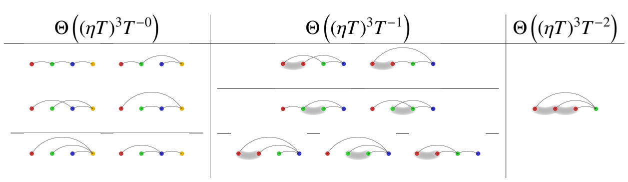


Table 1. Degree-3 diagrams for $B = M = 1$ SGD's test loss. The 6 diagrams have $(4 + 2) + (2 + 2 + 3) + (1)$ total orderings relevant to Proposition 1. **Left:** $(d, c) = (3, 0)$. Diagrams for ODE behavior. **Center:** $(d, c) = (3, 1)$. 1st order deviation of SGD away from ODE. **Right:** $(d, c) = (3, 2)$. 2nd order deviation of SGD from ODE with appearance of non-Gaussian statistics.

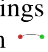
3.2. Exploiting curvature

Intuitively, the order-η^d truncation of Theorem 1's series depends on simple loss statistics near initialization, so it will fail when ηT is large enough for the weight to drift far from initialization. An especially interesting case where weights do not drift far is the case of SGD dynamics near an isolated minimum. A generic minimum is characterized among critical points by its curvature, so we analyze the case where H is positive. In doing so, we follow prior work that uses lower bounds on the loss landscape's curvature to restrict the hypothesis space to a small basin near a minimum and thus sharpen analyses of optimization and generalization (Bartlett et al., 2005).

We will incorporate the positive-H assumption into our theory via "re-summation" so that our re-summed order-η^d predictions near isolated minima will remain finite for fixed ηT and arbitrary T. More concretely, whenever we compute a diagram, we will also compute the unboundedly many cousins of that diagram that arise by inserting degree-2 nodes onto thin edges. We will sum these diagrams' contributions to Theorem 1's series, arriving at a closed form expression. Theorem 2 establishes the correctness of this approach. Thus, by thoroughly incorporating curvature information, re-summation will help us reason about long-term equilibrium near an isolated minimum and short-term drifts within a valley of minima.

To illustrate the idea, consider this class of topologically related diagrams: , , , $\cdots$. Intuitively, these diagrams all represent the effect of the leftmost node on the rightmost node, with some number of degree-2 nodes mediating. Since degree-2 nodes evaluate to hessians H, we regard these diagrams as versions of modulated by curvature. Each of the above diagrams has some number of embeddings into spacetime. Here (but not in Theorem 2), we will for simplicity consider embeddings into vanilla SGD's spacetime. Moreover, let us consider only embeddings that map the start and end nodes to fixed cells (n_0, t_0) and (n_+, t_+) separated $\Delta t = t_+ - t_0$ timesteps. We will also temporarily relax the constraint on embeddings by allowing each of the middle nodes to occupy any row — and in particular the same row as other nodes.* Then, a routine invocation of the Binomial Theorem shows that these embeddings together contribute the following to Theorem 1's series:

$$-G(I - \eta H)^{\Delta t - 1} \eta G$$

For comparison, the analogous embeddings (in this case, there is only one) of the smallest diagram  sum to

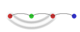
$$-G \eta G$$

* Because we here allow more embeddings than occur in Theorem 1, we are overcounting. It turns out that our use of differences as mentioned in Definition 6 leads to a telescoping cancellation that exactly counters this overcounting. We offer mathematical details in Appendix E.4 and the proof of Theorem 2. For now, we note that Theorem 2 will abstract away the middle nodes altogether, meaning that the problem of overcounting is relevant only to proof details.

which matches like the overall sum if we replace η by an "effective learning rate"

$$K(\Delta t) \triangleq (I - \eta H)^{\Delta t - 1}\eta$$

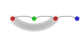
In the proof of Theorem 2, we see that this generalizes: in order to sum over a class of related diagrams' embeddings, we may sum over embeddings of the smallest diagram in that class, then replace each η corresponding to a duration-Δt edge by $K(\Delta t)$.

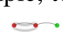
Example 4. The family , , , $\cdots$ includes variants of where we insert new nodes along 's two thin edges. The diagram evaluates to

$$\frac{1}{2}(\nabla_\mu C_{\nu\lambda})\eta^{\nu\lambda}\eta^{\mu\rho}G_\rho$$

So the overall family evaluates to

$$\frac{1}{2}(\nabla_\mu C_{\nu\lambda})K(\Delta t)^{\nu\lambda}K(\Delta t)^{\mu\rho}G_\rho$$

Definition 5. A diagram is *irreducible* when each of its degree-2 non-root nodes does not participate in fuzzy ties. So and but neither nor  are irreducible.

Definition 6 (Embedding-Sensitive Values)**.** Let $\text{rvalue}_f(D)$ be the expected value of D's corresponding tensor expression, where instead of using η to contract two tensors embedded to times $t, t+\Delta t$, we use $K(\Delta t) = (I - \eta H)^{\Delta t - 1}\eta$. Actually, it will be most convenient to let rvalues represent a *difference* from the noiseless case. For example, to compute rvalue(), we will replace η by $K(\Delta t)$ in ) instead of in). This way, each diagram represents a net effect of noise. For the small diagrams we consider, we obtain rvalues by replacing fuzzy ties by fuzzy outlines; larger diagrams present complications addressed in Appendix **??**.

Remark 1 (Re-summed Recipe)**.** In general, one sums over embeddings of irreducible diagrams, using $\text{rvalue}_f(D)$ instead of value(D). In practice, we approximate sums over embeddings by integrals over times and $(I - \eta H)^t$ by $\exp(-\eta H t)$, hence incurring a term-by-term multiplicative error of $1 + o(\eta)$ that preserves leading order results. Diagrams thus induce easily evaluated integrals of exponentials.

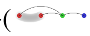
Theorem 2 (Re-summation Gives Large-T Limits)**.** *For any T: for η sufficiently small, SGD's expected test loss exceeds the noiseless case by*

$$\sum_{D \text{ irreducible}} \sum_{\text{embeddings } f} \frac{1}{|\text{Aut}_f(D)|} \frac{\text{rvalue}_f(D)}{(-B)^{|\text{edges}(D)|}}$$

As in Theorem 1: D ranges through diagrams whose root does not participate in any fuzzy ties, and f ranges through embedding of d. In contrast to Theorem 1: when H is positive, the dth order truncation converges as T diverges and ηT is fixed.

4. Insights from the Formalism

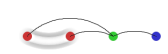
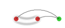
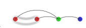
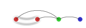
4.1. SGD descends on a C-smoothed landscape

Integrating rvalue_f() over embeddings f, we see:

Corollary 1 (Minima flat w.r.t. C attract SGD)**.** *Initialized at a test minimum, vanilla SGD's weight moves to order η^2 with a long-time-averaged[*] expected velocity of*

$$v^\pi = C_{\mu\nu}\left(F^{-1}\right)^{\mu\nu}_{\rho\lambda} J^{\rho\lambda}_\sigma \left(\frac{I - \exp(-T\eta H)}{T\eta H}\eta\right)^{\sigma\pi}$$

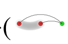
per timestep. Here, $F = \eta H \otimes I + I \otimes \eta H$, a 4-valent tensor.

The intuition behind the Corollary is that the diagram  contains a subdiagram  $= CH$; by a routine check, this subdiagram is the leading-order loss increase when we convolve the landscape with a C-shaped Gaussian. Since  connects the subdiagram to the test measurement via 1 edge, it couples to the linear part of the test loss and hence represents a displacement of weights away from high CH. In short,  reveals that *SGD descends on a covariance-smoothed landscape*. See Figure 2 (right).

An un-resummed version of this result was first reported by Yaida (2019b); however, for fixed T, the un-resummed result scales with η^3 while Corollary 1 scales with η^2. The discrepancy occurs, intuitively, because the re-summed analysis accounts for the accumulation of noise from many updates, hence amplifying the contribution of C. Our experiments verify our scaling law.

Unlike Wei & Schwab (2019), we make no assumptions of thermal equilibrium, fast-slow mode separation, or constant covariance. This generality reveals a novel dynamical phenomenon, namely that the velocity field above need not be conservative (see Section 5.4)

4.2. Curvature controls overfitting

Integrating rvalue_f() and rvalue_f() yields:

Corollary 2 (Flat, Sharp Minima Overfit Less)**.** *Initialized at a test minimum, pure GD's test loss is to order η*

$$\frac{1}{2N} C_{\mu\nu}\left((I - \exp(-\eta T H))^{\otimes 2}\right)^{\mu\nu}_{\rho\lambda}\left(H^{-1}\right)^{\rho\lambda}$$

above the minimum. This vanishes when H does. Likewise,

[*] That is, T so large that $C\exp(-\eta K T)$ is negligible. Appendix **??** gives a similar expression for general T.

pure GD's generalization gap is to order η:

$$\frac{1}{N} C_{\mu\nu} \left(I - \exp(-\eta T H)\right)^{\nu}_{\lambda} \left(H^{-1}\right)^{\lambda\mu}$$

In contrast to the later-mentioned Takeuchi estimate, this does not diverge as H shrinks.

Corollary 2's generalization gap converges after large T to $C_{\mu\nu}(H^{-1})^{\mu\nu}/N$, also known as Takeuchi's Information Criterion (TIC). In turn, $C = H$ is the Fisher metric in the classical setting of maximum likelihood (ML) estimation (in well-specified models) near the "true" test minimum, so we recover AIC (number of parameters)$/N$. Unlike AIC, our more general expression is descendably smooth, may be used with MAP or ELBO tasks instead of just ML, and makes no model well-specification assumptions.

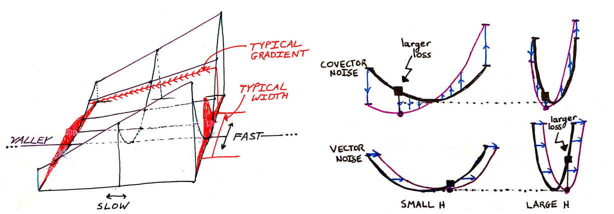
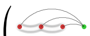
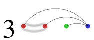


Figure 2. **Re-summation reveals novel phenomena. Left**: The entropic force mechanism: gradient noise induces a flow toward minima *with respect to to the covariance*. Though our analysis assumes neither thermal equilibrium nor fast-slow mode separation, we label "fast and slow directions" to ease comparison with Wei & Schwab (2019). Here, red densities denote the spread predicted by a re-summed $C^{\mu\nu}$, and the spatial variation of curvature corresponds to $J_{\mu\nu\lambda}$. **Right**: Noise structure determines how curvature affects overfitting. Geometrically, for (empirical risk minimization on) a vector-perturbed landscape, small Hessians are favored (top row), while for covector-perturbed landscapes, large Hessians are favored (bottom row). Corollary 2 shows how the implicit regularization of fixed-ηT descent interpolates between the two rows.

4.3. Nongaussian noise affects SGD and not ODE, SDE

Corollary 3 (SGD Differs from ODE and SDE)**.** *The test loss of vanilla SGD deviates at order T^{-1} from ODE by $\frac{T^2 T^{-1}}{2}$ $C_{\mu\nu} H^{\mu\nu}$. Its order T^{-2} deviation due to non-Gaussian noise is* $\frac{T^3 T^{-2}}{6}\Big($  $- 3$  $\Big) = -\frac{T^3 T^{-2}}{6}\left(\left(\mathbb{E}\left[\nabla_\mu l_x \nabla_\nu l_x \nabla_\lambda l_x\right] - G_\mu G_\nu G_\lambda\right) J^{\mu\nu\lambda} - 3 C_{\mu\nu} G_\lambda J^{\mu\nu\lambda}\right)$. *These effects contribute to SGD's difference from SDE.*

For finite N, these effects separate SDE from SGD. SDE also fails to model multi-epoch SGD's inter-update correlations. Conversely, as $N \to \infty$ so that SDE matches SGD, optimization and generalization respectively become computationally intractable and trivial and hence less interesting.

4.4. Effects of epoch number and batch size

Corollary 4 (Shuffling Barely Matters)**.** *To order η^3, inter-epoch shuffling doesn't affect SGD's expected test loss.*

Corollary 5 (The Effect of Epoch Number)**.** *To order η^2, one-epoch SGD has $\left(\frac{M-1}{M}\right)\left(\frac{B+1}{B}\right)\left(\frac{N}{2}\right)\left(\nabla_\mu C^\nu_\nu\right) G^\mu/2$ less test loss than M-epoch SGD with learning rate η/M.*

Analyzing , we find that we may cause GD to mimic SGD using any smooth unbiased estimator $\hat{C}$ of C:

Corollary 6 (The Effect of Batch Size)**.** *The expected test loss of pure SGD is, to order η^2, less than that of pure GD by $\frac{M(N-1)}{2}$ $\left(\nabla_\mu C^\nu_\nu\right) G^\mu/2$. Moreover, GD on a modified loss $\tilde{l}_n = l_n + \frac{N-1}{4N}$ $\hat{C}^\nu_\nu(\theta)$ has an expected test loss that agrees with SGD's to second order. We call this method GDC.*

5. Experiments

We focus on experiments whose rejection of the null hypothesis (and hence support of our theory) is so drastic as to be visually obvious. For example, in Figure 5, (Chaudhari & Soatto, 2018) predicts a velocity of 0 while we predict a velocity of $\eta^2/6$. Throughout, I bars and + marks denote a 95% confidence interval based on the standard error of the mean, in the vertical or vertical-and-horizontal directions, respectively. See Appendix **??** for experimental procedure including architectures and sample size.

5.1. Basic predictions

We test Theorem 1 on smooth convnets on CIFAR-10 and Fashion-MNIST. Our order η^3 predictions agree with experiment up to $\eta T \approx 10^0$ (Figure 3, left). Likewise, Corollary 5 correctly predicts the effect of multi-epoch training (Appendix **??**) for $\eta T \approx 10^{-1/2}$. These tests verify that our proofs hide no mistakes of proportionality or sign.

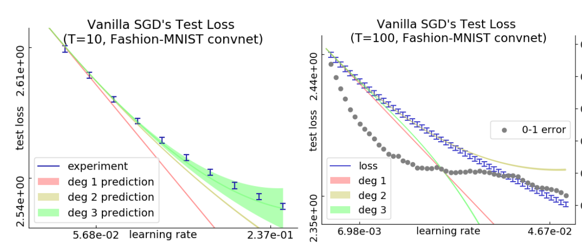


Figure 3. **Perturbation models SGD for small ηT.** Test loss vs learning rate on a Fashion-MNIST convnet, with un-re-summed predictions. **Left**: For the instance shown and all 11 other initializations unshown, our degree-3 prediction agrees with experiment through $\eta T \approx 10^0$, which corresponds to a decrease in 0-1 error of $\approx 10^{-3}$. **Right**: For larger ηT, our predictions can break down. Here, the order-3 prediction holds until the 0-1 error improves by $5 \cdot 10^{-3}$. Beyond this, close agreement with experiment is coincidental.

5.2. Emulating small batches with large ones

By Corollary 6, SGD avoids high-C regions more than GD We artificially correct GD accordingly, yielding an optimizer, GDC, that indeed behaves like SGD on a range of landscapes (Figure 4 (left)). It may be important to emulate SGD's avoidance of high-C regions because we C controls the rate at which each new update increases the generalization gap* (Figure 4 (right)).

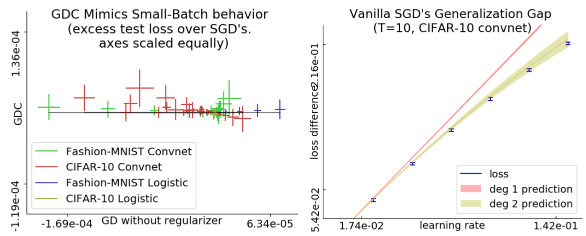


Figure 4. **C controls generalization and distinguishes GD from SGD. Left**: With equal-scaled axes, this plot shows that GDC matches SGD (small vertical variation) better than GD matches SGD (large horizontal variation) in test loss, for a variety of learning rates ($\approx 10^{-3} - 10^{-1}$) and initializations (zero and several Xavier-Glorot trials) on logistic and architectures for image classification. Here, $T = 10$. **Right**: CIFAR-10 generalization gaps. For the instance shown and all 11 other initializations unshown, the degree-2 prediction agrees with experiment through $\eta T \approx 5 \cdot 10^{-1}$.

The connection between generalization and covariance was first established by Roberts (2018) in the case $T = 2$ and to order η^2. In fact, that work conjectures the possibility of emulating GD with SGD. This sub-section extends that work by generalizing to arbitrary T and arbitrary orders η^d, and by concretely defining GDC.

In these experiments, we used a covariance estimator $\hat{C} \propto \nabla l_x(\nabla l_x - \nabla l_y)$ evaluated on two batches x, y that evenly partition the train set. For typical architectures, we may compute $\nabla \hat{C}$ with the same memory and time as the usual gradient ∇l_t, up to a multiplicative constant.

5.3. Comparison to continuous time

Consider fitting a centered normal $\mathcal{N}(0, \sigma^2)$ to some centered standard normal data. We parameterize the landscape by $h = \log(\sigma^2)$ so that the Fisher information matches the standard dot product (Amari, 1998). The gradient at sample x and weight σ is then $g_x(h) = (1 - x^2 \exp(-h))/2$. Since $x \sim \mathcal{N}(0, 1)$, $g_x(h)$ will be affinely related to a chi-squared, and in particular non-Gaussian. At $h = 0$, the expected gradient vanishes, and the test loss of vanilla SGD only involves diagrams with no singleton leaves; to third order, it is $\cdot + \frac{T}{2} + \binom{T}{2} + \frac{T}{6}$ In particular, the $\binom{T}{2}$ differs from $T^2/2$ and hence contributes to the time-discretization error of SDE as an approximation for SDE. Moreover, non-Gaussian noise contributes via to that error. Appendix **??** shows that SDE and one-epoch SGD indeed differ. For multi-epoch SGD, the effect of overfitting to finite training data further separates SDE and SGD.

*Reminder: for us, generalization gap is test minus train loss.

5.4. Nonconservative entropic force

To test Corollary 1's predicted force, we construct a counter-intuitive loss landscape wherein, for arbitrarily small learning rates, SGD steadily increases the weight's z component despite 0 test gradient in that direction. Our mechanism differs from that discovered by Chaudhari & Soatto (2018). Specifically, because in this landscape the force is η-perpendicular to the image of ηC, that work predicts an entropic force of 0. This disagreement in predictions is possible because our analysis does not make any assumptions of equilibrium, conservatism, or continuous time.

So, even in a valley of global minima, SGD will move away from minima whose Hessian aligns with the current covariance. However, by the time it moves, the new covariance might differ from the old one, and SGD will be repelled by different Hessians than before. Setting the covariance to lag the Hessian by a phase, we construct a landscape in which this entropic force dominates. This "*linear screw*" landscape has 3-dimensional $w \in \mathbb{R}^3$ (initialized to 0) and 1-dimensional $x \sim \mathcal{N}(0, 1)$:

$$l_x(w) \triangleq \frac{1}{2} H(z)(w, w) + x \cdot S(z)(w)$$

Here, $H(z)(w, w) = w_x^2 + w_y^2 + (\cos(z) w_x + \sin(z) w_y)^2$ and $S(z)(w) = \cos(z - \pi/4) w_x + \sin(z - \pi/4) w_y$. There is a valley of global minima defined by $x = y = 0$. If SGD is initialized there, then to leading order in η and for large T, the re-summed theory predicts a z-speed of $\eta^2/6$ per timestep. Our re-summed predictions agree for with experiment for ηT so large that the weight moves about 5 times the landscape's natural length scale of 2π (Figure 5, left).

It is routine to check that, by stitching together copies of this example, we may cause SGD to travel along paths that are closed loops or unbounded curves. We may even add a small linear component so that SGD steadily climbs uphill.

5.5. Sharp and flat minima both overfit less

Prior work has varyingly found that *sharp* minima overfit less (after all, l^2 regularization increases curvature) or that *flat* minima overfit less (after all, flat minima are more robust to small displacements in weight space). Corollary 2 reconciles these competing intuitions by showing how the relationship of generalization and curvature depends on the learning task's noise structure and how the metric η^{-1}

mediates this distinction (Figure 2, right).

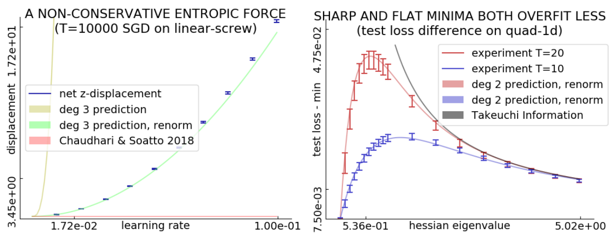


Figure 5. **Re-summed predictions excel even for large ηT for SGD near minima. Left**: On Linear Screw, the persistent entropic force pushes the weight through a valley of global minima not at a $T^{1/2}$ diffusive rate but at a directional T^1 rate. Since Hessians and covariances are bounded throughout the valley and the effect appears for all sufficiently small η, the effect is not a pathological artifact of well-chosen learning rate or divergent covariance noise. The net displacement of $\approx 10^{1.5}$ well exceeds the z-period of 2π. **Right**: For Mean Estimation with fixed covariance and a range of Hessians, initialized at the true minimum, the test losses after fixed-ηT optimization are smallest for very small and very large curvatures. This evidences our prediction that both sharp and flat minima overfit less and that TIC's singularity is suppressed.

Because the TIC estimates a smooth hypothesis class's generalization gap, it is tempting to use it as an additive regularization term. However, since the TIC is singular where the Hessian is singular, it gives insensible results for overparameterized models. Indeed, Dixon & Ward (2018) reports numerical difficulties requiring an arbitrary cutoff.

Fortunately, by Corollary 2, the implicit regularization of gradient descent both demands and enables a singularity-removing correction to the TIC (Figure 5, right). The resulting *Stabilized TIC* (STIC) uses the metric η^{-1} implicit in gradient descent to threshold flat from sharp minima[*]. It thus offers a principled method for optimizer-aware model selection easily compatible with automatic differentiation systems. By descending on STIC, we may tune smooth hyperparameters such as l_2 coefficients. Experiments on an artificial Mean Estimation problem (task in Appendix **??**, plot in Appendix **??**) recommend STIC for model selection when C/N dwarves H as in the noisy, small-N regime. Because diagonalization typically takes time cubic in dimension, exact STIC regularization is most useful for small models on noisy and limited data.

6. Related Work

It was Kiefer & Wolfowitz (1952) who, in uniting gradient descent (Cauchy, 1847) with stochastic approximation (Robbins & Monro, 1951), invented SGD. Since the development of back-propagation for efficient differentiation (Werbos, 1974), SGD has been used to train connectionist models including neural networks (Bottou, 1991), in recent years to remarkable success (LeCun et al., 2015).

[*] The notion of H's width depends on a choice of metric. Prior work chooses this metric arbitrarily. We show that choosing η^{-1} is a natural choice because it leads to a prediction of the generalization gap.

Several lines of work quantify the overfitting of SGD-trained networks (Neyshabur et al., 2017a). For instance, Bartlett et al. (2017) controls the Rademacher complexity of deep hypothesis classes, leading to generalization bounds that are optimizer-agnostic. However, since SGD-trained networks generalize despite their seeming ability to shatter large sets (Zhang et al., 2017), one infers that generalization arises from the aptness to data of not only architecture but also optimization (Neyshabur et al., 2017b). Others have focused on the implicit regularization of SGD itself, for instance by modeling descent via stochastic differential equations (SDEs) (e.g. Chaudhari & Soatto (2018)). However, per Yaida (2019a), such continuous-time analyses cannot treat covariance correctly, and so they err when interpreting results about SDEs as results about SGD for finite trainsets.

Following Roberts (2018), we avoid continuous-time approximations and Taylor-expand around $\eta = 0$. We hence extend that work beyond leading order and beyond 2 time steps, allowing us to compare, for instance, the expected test losses of multi-epoch and one-epoch SGD. We also quantify the overfitting effects of batch size, whence we propose a regularizer that causes large-batch GD to emulate small-batch SGD. In doing so, we establish a precise version of the relationship — between covariance, batch size, and generalization — conjectured by Jastrzębski et al. (2018).

While we make rigorous, architecture-agnostic predictions of learning curves, these predictions become vacuous for large η. Other discrete-time dynamical analyses allow large η by treating deep generalization phenomenologically, whether by fitting to an empirically-determined correlate of Rademacher bounds (Liao et al., 2018), by exhibiting generalization of local minima *flat* with respect to the standard metric (see Hoffer et al. (2017), Keskar et al. (2017), Wang et al. (2018)), or by exhibiting generalization of local minima *sharp* with respect to the standard metric (see Stein (1956), Dinh et al. (2017), Wu et al. (2018)). Our work reconciles those seemingly clashing claims.

Others have perturbatively analyzed descent: Dyer & Gur-Ari (2019) perturb in inverse network width, employing Feynman-'t Hooft diagrams to correct the Gaussian Process approximation for a specific class of deep networks. Meanwhile, (Chaudhari & Soatto, 2018) and Li et al. (2017) perturb in learning rate to second order by approximating noise between updates as Gaussian and uncorrelated. In neglecting correlations and heavy tails, that work neither extends to higher orders not describes SGD's generalization behavior. By contrast, we use Feynman-Penrose diagrams to compute test and train losses to arbitrary order in learning rate. Our method accounts for non-Gaussian and correlated noise and applies to *any* sufficiently smooth architecture. For example, since our work does not rely on information-geometric relationships between C and H (Amari, 1998)[*], it applies to inexact-likelihood landscapes such as VAEs'.

7. Conclusion

We presented an elegant diagram-based framework for studying short-time SGD. Theorem 2 justifies long-time predictions of SGD's dynamics near minima. Our theory answers the following questions.

Which Minima Overfit Less? By analyzing , we find that flat and sharp minima both overfit less than minima of curvature comparable to $(\eta T)^{-1}$. Flat minima are robust to vector-valued noise, sharp minima are robust to covector-valued noise, and medium minima attain the worst of both worlds. We thus reconcile prior intuitions that sharp (Keskar et al., 2017; Wang et al., 2018) or flat (Dinh et al., 2017; Wu et al., 2018) minima overfit worse. These considerations lead us to a smooth generalization of AIC enabling hyperparameter tuning by gradient descent.

Which Minima Does SGD Prefer? Analyzing , we refine Wei & Schwab (2019) to nonconstant, nonisotropic covariance to reveal that SGD descends on a loss landscape smoothed by the *current* covariance C. In particular, SGD moves toward regions flat with respect to C. As C evolves, the smoothing mask and thus the effective landscape evolves. This dynamics is generically nonconservative. In contrast to Chaudhari & Soatto (2018)'s SDE approximation, SGD does not generically converge to a limit cycle.

Can GD Emulate SGD? By analyzing , we prove the conjecture of Roberts (2018), that large-batch GD can be made to emulate small-batch SGD. We show how to do this by adding a multiple of an unbiased covariance estimator to the descent objective. This emulation is significant because, while small batch sizes can lead to better generalization (Bottou, 1991), modern infrastructure increasingly rewards large batch sizes (Goyal et al., 2018).

7.1. Consequences

Our analysis of which minima (among a valley of minima) SGD prefers, — and our characterization of when SGD overfits less in certain minima — together offer insight into SGD's success in training over-parameterized models.

Our results may also help to analyze fine-tuning procedures such as the meta-learning of MAML ((Finn et al., 2017)). Indeed, those methods seek models initialized near minima and tunable to new data through a small number of updates, a setting matched to our theory's assumptions.

Since our predictions depend only on loss data near initialization, they break down after the weight moves far from initialization. Our theory thus best applies to small-movement contexts, whether for long times (large ηT) near an isolated minimum or for short times (small ηT) in general.

Yet, even short-time predictions show how curvature and noise — and not just averaged gradients — repel or attract SGD's current weight. For example, we proved that SGD in a valley moves toward regions flat with respect to the current covariance C. Initial data rarely suffices to predict long-time behavior because landscapes can be arbitrarily complex. Much as meteorologists understand how warm and cold fronts interact despite the intractability of long-term weather forecasting, our contribution is to quantify the counter-intuitive dynamics governing SGD's short-time behavior.[†] Our results enhance intuitions relied on by practitioners — e.g. that "SGD descends on the train loss" — by summarizing the effect of noise in closed-form dynamical laws valid in each short-term patch of SGD's trajectory.

7.2. Questions

The diagram method opens the door to exploration of Lagrangian formalisms and curved backgrounds[‡]:

Question 1. *Does some least-action principle govern SGD; if not, what is an essential obstacle to this characterization?*

Lagrange's least-action formalism intimately intertwines with the diagrams of physics. Together, they afford a modular framework for introducing new interactions as new terms or diagram nodes. In fact, we find that some *higher-order* methods — such as the Hessian-based update $\theta \leftsquigarrow \theta - (\eta^{-1} + \lambda\nabla\nabla l_t(\theta))^{-1}\nabla l_t(\theta)$ parameterized by small η, λ — admit diagrammatic analysis when we represent the λ term as a second type of diagram node. Though diagrams suffice for computation, it is Lagrangians that most deeply illuminate scaling and conservation laws.

Conjecture 1 (Riemann Curvature Regularizes)**.** *For small η, SGD's gen. gap decreases as sectional curvature grows.*

Though our work so far assumes a flat metric $\eta^{\mu\nu}$, it generalizes to curved weight spaces[§]. Curvature finds concrete application in the *learning on manifolds* paradigm of Absil et al. (2007); Zhang et al. (2016), notably specialized to Amari (1998)'s *natural gradient descent* and Nickel & Kiela (2017)'s *hyperbolic embeddings*. We are optimistic

[*] Disagreement of C and H is typical in modern learning (Roux et al., 2012; Kunstner et al., 2019).

[†] Because our analysis holds for any initialization, one may imagine SGD's coarse-grained trajectory as an integral curve of the vector field given by our theory.

[‡] Landau and Lifshitz introduce these concepts (1960; 1951).

[§] One may represent the affine connection as a node, thus giving rise to non-tensorial and hence gauge-dependent diagrams.

our formalism may resolve conjectures such as above.

References

Absil, P.-A., Mahony, R., and Sepulchre, R. Optimization algorithms on matrix manifolds, chapter 4. *Princeton University Press*, 2007.

Amari, S.-I. Natural gradient works efficiently. *Neural Computation*, 1998.

Bartlett, P., Bousquet, O., and Mendelson, S. Local rademacher complexities. *Annals of Statistics*, 2005.

Bartlett, P., Foster, D., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. *NeurIPS*, 2017.

Bonnabel, S. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 2013.

Bottou, L. Stochastic gradient learning in neural networks. *Neuro-Nîmes*, 1991.

Cauchy, A.-L. Méthode générale pour la résolution des systémes d'équations simultanées. *Comptes rendus de l'Académie des Sciences*, 1847.

Chaudhari, P. and Soatto, S. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *ICLR*, 2018.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. *ICLR*, 2017.

Dixon, M. and Ward, T. Takeuchi's information criteria as a form of regularization. *Arxiv Preprint*, 2018.

Dyer, E. and Gur-Ari, G. Asymptotics of wide networks from feynman diagrams. *ICML Workshop*, 2019.

Feynman, R. A space-time appxoach to quantum electrodynamics. *Physical Review*, 1949.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 2017.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd. *Data @ Scale*, 2018.

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better. *NeurIPS*, 2017.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in sgd. *Arxiv Preprint*, 2018.

Keskar, N., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017.

Kiefer, J. and Wolfowitz, J. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 1952.

Kunstner, F., Hennig, P., and Balles, L. Limitations of the empirical fisher approximation for natural gradient descent. *NeurIPS*, 2019.

Landau, L. and Lifshitz, E. The classical theory of fields. *Addison-Wesley*, 1951.

Landau, L. and Lifshitz, E. Mechanics. *Pergamon Press*, 1960.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 2015.

Li, Q., Tai, C., and E, W. Stochastic modified equations and adaptive stochastic gradient algorithms i. *PMLR*, 2017.

Liao, Q., Miranda, B., Banburski, A., Hidary, J., and Poggio, T. A surprising linear relationship predicts test performance in deep networks. *Center for Brains, Minds, and Machines Memo 91*, 2018.

Nesterov, Y. Lectures on convex optimization: Minimization of smooth functions. *Springer Applied Optimization 87, Section 2.1*, 2004.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. *NeurIPS*, 2017a.

Neyshabur, B., Tomioka, R., Salakhutdinov, R., and Srebro, N. Geometry of optimization and implicit regularization in deep learning. *Chapter 4 from Intel CRI-CI: Why and When Deep Learning Works Compendium*, 2017b.

Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. *ICML*, 2017.

Penrose, R. Applications of negative dimensional tensors. *Combinatorial Mathematics and its Applications*, 1971.

Robbins, H. and Monro, S. A stochastic approximation method. *Pages 400-407 of The Annals of Mathematical Statistics.*, 1951.

Roberts, D. Sgd implicitly regularizes generalization error. *NeurIPS: Integration of Deep Learning Theories Workshop*, 2018.

Roux, N., Bengio, Y., and Fitzgibbon, A. Improving first and second-order methods by modeling uncertainty. *Book Chapter: Optimization for Machine Learning, Chapter 15*, 2012.

Stein, C. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. *Berkeley Symposium on Mathematical Probability*, 1956.

Wang, H., Keskar, N., Xiong, C., and Socher, R. Identifying generalization properties in neural networks. *Arxiv Preprint*, 2018.

Wei, M. and Schwab, D. How noise affects the hessian spectrum in overparameterized neural networks. *Arxiv Preprint*, 2019.

Werbos, P. Beyond regression: New tools for prediction and analysis. *Harvard Thesis*, 1974.

Wu, L., Ma, C., and E, W. How sgd selects the global minima in over-parameterized learning. *NeurIPS*, 2018.

Yaida, S. Fluctuation-dissipation relations for stochastic gradient descent. *ICLR*, 2019a.

Yaida, S. A first law of thermodynamics for stochastic gradient descent. *Personal Communication*, 2019b.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ICLR*, 2017.

Zhang, H., Reddi, S., and Sra, S. Fast stochastic optimization on riemannian manifolds. *NeurIPS*, 2016.

Organization of Appendices

These three appendices respectively serve three functions:

- to explain how to calculate using diagrams;
- to prove our theorems, corollaries, and claims; and
- to specify our experimental methods and results.

A. How to Calculate Expected Test Losses Using Diagrams

Our work introduces a novel technique for calculating the expected learning curves of SGD in terms of statistics of the loss landscape near initialization. Here, we explain this technique. There are **four steps** to computing the expected test loss after a specific number of gradient updates:

- Based on the chosen optimization hyperparameters (namely, batch size, training set size, and number of epochs): **draw the spacetime grid** that encodes these hyperparameters.
- Based on our desired level of precision, **draw all the relevant embeddings** of diagrams into the spacetime.
- **Evaluate each diagram embedding**.
- **Sum the embeddings' values** to obtain the quantity of interest as a function of the learning rate.

After presenting a small, complete example calculation that follows these four steps, we explain how to perform each of these steps in its own sub-section. We then discuss how diagrams often offer intuition as well as calculational help. Though we focus on the computation of expected test losses, we explain how a small change in the above four steps allows for the computation also of variances (instead of expectations) and of train losses (instead of test losses). We conclude by comparing direct calculation based on our Key Lemma to the diagram method; we point out when and why diagrams streamline computation.

A.1. An example calculation

Let's compute the expected test loss of batchsize-1 SGD on N training points after E epochs. We'll do this calculation to order η^2, meaning that our answer will be a function of the learning rate η and that its error will shrink faster than quadratically as η becomes small.

First, we identify the relevant spacetime. A spacetime is a set of cells indicating which training points are used in which gradient update. Since our problem has NE many updates, each on a batch of size 1, there will be $NE \times 1$ many cells in the relevant spacetime. We arrange these cells in a grid whose vertical axis indexes training points and whose horizontal axis indexes training times: FILL IN

Next, we identify the relevant diagram embeddings. The benefit of drawing this diagram.

Then, we evaluate each diagram embedding. The benefit of drawing this diagram.

Finally, we sum the embeddings' values to arrive at an answer. The benefit of drawing this diagram.

A.2. How to identify the relevant space-time

A.3. How to identify the relevant diagram embeddings

A.4. How to evaluate each embedding

A.5. How to sum the embeddings' values

A.6. Interpreting diagrams to build intuition

A.7. How to solve variant problems

A.8. Do diagrams streamline computation?

B. Assumptions and Proofs

B.1. Setup and assumptions of our theory

B.2. Proof of the Key Lemma

B.3. Proof of Theorem 1

B.4. On Mobius inversion for resummed diagrams

B.5. Proof of Theorem 2

B.6. Proofs of corollaries

B.7. Proofs of miscellaneous claims

C. Exerimental Methods and Results

C.1. What loss landscapes did we use?

C.2. Implementing optimizers

C.3. Evaluating diagrams on a given landscape

C.4. Software frameworks and hardware

C.5. Additional figures