Proceedings of Machine Learning Research vol 134:1–48, 2021 34rd Annual Conference on Learning Theory

Perturbation Theory of Stochastic Gradient Descent

author names withheld

Editor: Under Review for COLT 2021

Abstract

We quantify how gradient noise shapes the dynamics of stochastic gradient descent (SGD) by taking Taylor series in the learning rate. We present in particular a new diagram-based notation that permits resummation to convergent results. We employ our theory to contrast SGD against two popular approximations: deterministic descent and stochastic differential equations. We find that SGD's trajectory avoids regions of weight space with high gradient noise and avoids minima that are sharp with respect to gradient noise.

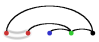
Keywords: SGD, learning rates, generalization, gradient noise, perturbation.

1. Introduction

Gradient estimates, measured on minibatches and thus noisy, form the primary learning signal when training deep neural nets. While users of deep learning benefit from the intuition that such *stochastic gradient descent* (SGD) approximates deterministic gradient descent (GD) (Bottou, 1991; LeCun et al., 2015), SGD's gradient noise in practice alters training dynamics and testing losses (Goyal et al., 2018; Wu et al., 2020). This paper studies SGD on short timescales or near minima and shows that **gradient noise biases learning** toward low-curvature, low-noise regions of the loss landscape.

Generalizing Liao et al. (2018); Wei and Schwab (2019); Zhu et al. (2019), we model correlated, non-gaussian, non-isotropic, non-constant gradient noise and find qualitative differences in dynamics. For example, we construct a non-pathological[1] loss landscape on which SGD's trajectory *ascends*. We verify our theory on convolutional CIFAR-10 and Fashion-MNIST loss landscapes.

Our theory offers a new physics-inspired perspective of SGD as a superposition of concurrent information-flow processes. Indeed, we study the post-training testing loss by Taylor expanding it w.r.t. the learning rate η. We interpret the resulting terms as describing processes by which data influence weights. E.g. an instance of the process[2]



is shown on the right. Notating processes with such diagrams, we show in §2.4 how to compute the effect of each process and that summing the finitely many processes with d or fewer edges suffices to answer dynamical questions to error $o(\eta^d)$. We thus factor the analysis of SGD into the analyses of individual processes, a technique that may power future theoretical inquiries.

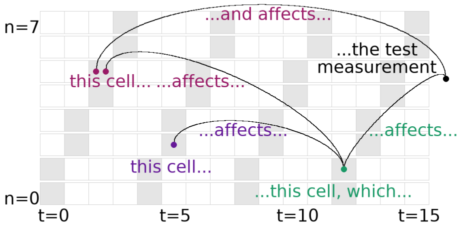


Figure 1: **A sub-process of SGD**. Timesteps index columns; training data index rows. The 5th datum participates in the 2nd SGD update. This $(n = 5, t = 2)$ event affects the testing loss both directly and via the $(1, 12)$ event, which is also modulated by the $(2, 5)$ event.

[1] All higher derivatives exist and are quadratically bounded; the gradient noise at each weight vector is 1-subgaussian.

[2] Throughout, colors help us refer to parts of diagrams; colors lack mathematical meaning.

© 2021 .

1.1. Background, notation, assumptions

We formalize the loss — suffered by a fixed architecture on a random datapoint — as a distribution $\mathcal{D}$ over functions from a weight-space $\mathcal{M}$. The *testing loss* $l : \mathcal{M} \to \mathbb{R}$ is $\mathcal{D}$'s mean. We write $\theta \in \mathcal{M}$, $l_x \sim \mathcal{D}$ for generic elements. We consider training sequences $(l_n : 0 \leq n < N) \sim \mathcal{D}^N$. We refer to n and to l_n as *training points*. We assume §B.1's hypotheses, e.g. that l, l_x are analytic and that all moments exist. Our theory accounts for tanh (but not ReLU) networks with cross entropy loss on bounded data — with weight sharing, skip connections, soft attention, and dropout.

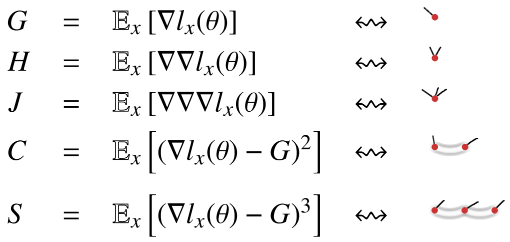
SGD performs η-steepest descent on the estimates l_n. Our theory describes SGD with any number **N** of training points, **T** of updates, and **B** of points per batch. Specifically, SGD runs T many updates (hence $\mathbf{E} = TB/N$ epochs or $\mathbf{M} = T/N$ updates per training point) of the form

$$\theta^\mu := \theta^\mu - \eta^{\mu\nu} \nabla_\nu \textstyle\sum_{n \in \mathcal{B}_t} l_n(\theta)/B$$

where in each epoch, we sample the tth batch $\mathcal{B}_t$ without replacement from the training sequence. So each initialization $\theta_0 \in \mathcal{M}$ induces a distribution over trajectories $(\theta_t : 0 \leq t \leq T)$, with randomness due both to training data and batch selection. We especially study the **final testing loss** $\mathbb{E}[l(\theta_T)]$.

We make heavy use Figure 2's tensors; G, H, J, C, S have $1, 2, 3, 2, 3$ (lower) indices, respectively. We implicitly sum repeated Greek indices: if a covector U and a vector V[1] have coefficients U_μ, V^μ, then $U_\mu V^\mu \triangleq \sum_\mu U_\mu \cdot V^\mu$. We regard the learning rate as an inverse metric $\eta^{\mu\nu}$ that converts gradient covectors to displacement vectors (Bonnabel, 2013). We use the learning rate η to raise indices; thus, $H^\mu_\lambda \triangleq \sum_\nu \eta^{\mu\nu} H_{\nu\lambda}$ and $C^\mu_\mu \triangleq \sum_{\mu\nu} \eta^{\mu\nu} \cdot C_{\nu\mu}$.

Little-os by default indicate $\eta \to 0$ limits potentially non-uniform in parameters (such as T) other than η. "*Order d*" means *with error term* $o(\eta^d)$ and "*leading order*" means *with error factor* $1 + o(1)$.



$G = \mathbb{E}_x[\nabla l_x(\theta)]$ ⇝

$H = \mathbb{E}_x[\nabla\nabla l_x(\theta)]$ ⇝

$J = \mathbb{E}_x[\nabla\nabla\nabla l_x(\theta)]$ ⇝

$C = \mathbb{E}_x\left[(\nabla l_x(\theta) - G)^2\right]$ ⇝

$S = \mathbb{E}_x\left[(\nabla l_x(\theta) - G)^3\right]$ ⇝

Figure 2: **Named tensors**, typically evaluated at initialization ($\theta = \theta_0$). §2.2 explains how diagrams depict tensors.

To illustrate our notation, we quote a well-known proposition (Nesterov (2004), §2.1):

Proposition 0 *G controls the leading order loss decrease:* $\mathbb{E}[l(\theta_T) - l(\theta_0)] = -TG_\mu G^\mu + o(\eta^1)$.

One proves this estimate by induction on T. When the loss landscape is noiseless and linear (that is, when $\nabla l_x(\theta)$ depends on neither x nor θ), this estimate is exact.

This paper's contributions are two-fold: first, to identify how gradient noise and curvature correct Proposition 0; and second, to replace induction by less opaque and more convergent large-T techniques. For example, our framework allows us to assess how gradient noise's non-Gaussianity affects the final testing loss. §2 details how evaluation of a *single diagram*

gives a case of Corollary 3, quoted here assuming isotropic curvature ($\eta H \propto I$) and $E = B = 1$:

Proposition 1 *If we initialize near a non-degenerate minimum of l, then in the large-T limit, the skewness S of gradient noise contributes* $-S_{\alpha\beta\gamma} J^{\alpha\beta\gamma}/18\|\eta H\|_2 + o(\eta^2)$ *to the final testing loss.*

So skewness affects loss in proportion to the logarithmic derivative J/H of curvature. The dependence on η is second order[2] and is hence a leading correction to Proposition 0. Gaussian approximations (e.g. SDE) miss this effect.

[1] Vectors/covectors, a.k.a. column/row vectors, represent distinct geometric concepts (Kolář et al., 1993).

[2] Three ηs raise J's indices; one η appears in the denominator $18\|\eta H\|$.

2. Perturbative theory of SGD

§2.1 illustrates by example a Taylor series approach to studying SGD dynamics. §2.2 introduces a diagram notation for the formulae that thus appear. §2.3 shows how to evaluate diagrams to obtain numbers. §2.4 states our main result: that diagram-based analysis is correct.

2.1. Challenges when using Taylor series to study SGD

We analyze Proposition 0 in detail to highlight our points of departure. This paradigmatic example demonstrates how Taylor expanding dynamical quantities of interest with respect to η results in a sum over disjoint processes, a separation mentioned on page 1. It also illustrates two obstacles, addressed in in §2.3, to Taylor-based analysis: there is a combinatorial explosion of terms, and any truncation of the Taylor series leads to a divergent result for large T.

We justify Proposition 0 by showing that $\nabla l_x(\theta)$'s dependence on x (due to gradient noise) and on θ (due to l's curvature) is irrelevant to first order. Then each step of SGD intuitively displaces θ by $-\eta G$. So T steps displace θ by $-T\eta G$. Since l rises by Gv for each displacement v, the overall change in l is $-GT\eta G$. We follow Nesterov (2004); Roberts (2018) to rigorize this intuition:

Proof (of Proposition 0). By the smoothness assumptions of §B.1, we have $\theta^\mu_T - \theta^\mu_0 \in O(\eta^1)$ for each T. We claim that $\theta^\mu_T - \theta^\mu_0 = -T\eta^{\mu\nu}G_\nu + o(\eta^1)$. The claim holds when SGD is run for $T = 0$ timesteps. Moreover, if $T = T' + 1$ and the claim holds when SGD is run for T' timesteps, then:

$$\begin{aligned}
\theta_T - \theta_{T'} &= -\eta\nabla l_{T'}(\theta_{T'}) \\
&= -\eta\nabla(l_{T'}(\theta_0) + \nabla l_{T'}(\theta_0)\cdot(\theta_{T'} - \theta_0) + o(\theta_{T'} - \theta_0)) \\
&= -\eta\nabla(l_{T'}(\theta_0) + \nabla l_{T'}(\theta_0)\cdot O(\eta^1) + o(O(\eta^1))) \\
&= -\eta\nabla l_{T'}(\theta_0) + o(\eta^1)
\end{aligned}$$

Here, we write $l_{T'}$ as shorthand for the batch average $\sum_{n\in\mathcal{B}_{T'}} l_n(\theta)/B$. Applying the induction hypothesis proves the claim. Finally, we plug the claim into a Taylor expansion of l:

$$\begin{aligned}
\mathbb{E}[l(\theta_T) - l(\theta_0)] &= \nabla l(\theta_0)\cdot\mathbb{E}[\theta_T - \theta_0] + \mathbb{E}[o(\theta_T - \theta_0)] \\
&= \nabla l(\theta_0)\cdot(-T\eta G + o(\eta^1)) + o(O(\eta^1)) \\
&= -T^1 G\eta G + o(\eta^1)\square
\end{aligned}$$

Indeed, by the moment assumptions of §B.1, the above expectations of $o(\eta^1)$ terms are still $o(\eta^1)$. ■

This paper extends the above by keeping higher order terms. The above shares with its naïve generalizations these three features, discussed below in detail: each update contributes *separately* to the loss; the number of intermediate terms *explodes* as T grows; the result is a *polynomial* in T.

By **separation**, we mean that to first order the tth update contributes the same amount (namely, $-\eta\mathbb{E}[\nabla l_t(\theta_0)]$) in expectation to θ_T no matter the value of $T > t$ (indeed, in the above, $-\eta\nabla l_{T'}(\theta_0)$ is constant with respect to θ_t for $t > 0$). In other words, to first order, future updates do not modulate the effect of past updates. We may thus imagine each update as independently affecting the final test loss. This separation may be surprising, for it fails when we expand to higher order and hence account for curvature: SGD on a bounded landscape does not descend indefinitely at a constant rate. Still, for higher orders of expansion there is an analogue of separaion that finds precise expression in Theorem 1. An intuition arises of many concurrent processes, each blind to all but d updates.

Expanding to higher order, we encounter an **explosion** of terms. To order η^2, for instance, we may not discard the hessian term $\eta\nabla\nabla l_{T'}(\theta_0)\cdot(\theta_{T'}-\theta_0)$. The order η^1 contributions of all updates prior to T' thus each contribute via $\theta_{T'}$ to this undiscarded term. So, while the proof above sums $O(T)$ terms, an order η^2 analysis sums $O(T^2)$ terms, each (e.g. $\nabla\nabla l_{t=5}\nabla l_{t=2}$) involving a *pair* of times. The terms vary in form, too: some (perhaps $\nabla\nabla l_{t=5}\nabla l_{t=2}$) of the quadratic terms that replace $\mathbb{E}[\theta_T-\theta_0]$ have statistically independent factors that permit expectations to factor, while others (perhaps $\nabla\nabla l_{t=5}\nabla l_{t=5}$) do not. This is how covariances of noise appear in our analysis.

A routine induction on (d,T) characterizes this explosion: for each d, there are $O(T^d)$ many terms when we expand to order η^d. Thus the result most analogous to Proposition 0's result estimates the expected final testing loss as a degree-d **polynomial** in T. For instance, $-T^1G\eta G$ is degree 1 in T. Though polynomial dependence is correct for fixed T and η sufficiently small relative to T, it is absurd when we fix η and vary T:[1] for example, SGD on a bounded loss function should not lead to unbounded loss as T grows. By contrast, our Theorems 1, 2 produce expressions that for each T have the correct Taylor data with respect to η and that for each η are bounded as T grows. We achieve this result by cancelling the aforementioned polynomial divergences at each order with higher order terms. We perform this **resummation** without omission or redundancy so that absolute convergence ensures consistency with the un-resummed result. What we find is that a term that involves updates separated in time decays exponentially in the separation.

2.2. Diagrams arise from Taylor series and depict information-flow processes

This section prepares for §2.3, which uses diagrams to generalize §2.1's separation, tame §2.1's combinatorial explosion of terms, and temper §2.1's polynomial divergence. We start by characterizing the higher order terms. Suppose s is an analytic function on $\mathcal{M}$. For example, s might be the testing loss l. The following Lemma, reminiscent of Dyson (1949)'s, tracks $s(\theta)$ as SGD updates θ:

Key Lemma *For all T: for η sufficiently small, $s(\theta_T)$ is a sum over tuples of natural numbers:*

$$\sum_{(d_t:0\leq t<T)\in\mathbb{N}^T}(-\eta)^{\sum_t d_t}\left(\prod_{0\leq t<T}\left(\left.\frac{(g\nabla)^{d_t}}{d_t!}\right|_{g=\sum_{n\in\mathcal{B}_t}\nabla l_n(\theta)/B}\right)\right)(s)(\theta_0) \tag{1}$$

Moreover, an expectation symbol (over training sets) commutes with the outer sum.

Here, we consider each $(g\nabla)^{d_t}$ as a higher order function that takes in a function f defined on weight space and outputs a function equal to the d_tth derivative of f, times g^{d_t}. The above product then indicates composition of $(g\nabla)^{d_t}$'s across the different t's. In total, that product takes the function s as input and outputs a function equal to some polynomial of s's derivatives.

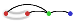
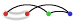
For example, the η^3 terms that appear in the above (for $s=l$) include:

$$-\nabla_\mu l_{t=2}\nabla_\nu l_{t=2}\nabla^\mu\nabla^\nu\nabla_\lambda l_{t=5}\nabla^\lambda l \qquad\qquad -\nabla_\mu l_{t=2}\nabla_\lambda l_{t=2}\nabla^\mu\nabla_\nu l_{t=5}\nabla^\nu\nabla^\lambda l$$

Let us take expectations over training sets. Suppose $B=1$ and $N>5$ so that the batches at $t=2,5$ are statistically independent. Then the expectations factor as:

$$-\mathbb{E}[\nabla_\mu l_{t=2}\nabla_\nu l_{t=2}]\mathbb{E}[\nabla^\mu\nabla^\nu\nabla_\lambda l_{t=5}]\mathbb{E}[\nabla^\lambda l] \qquad\qquad -\mathbb{E}[\nabla_\mu l_{t=2}\nabla_\nu l_{t=2}]\mathbb{E}[\nabla^\mu\nabla_\lambda l_{t=5}]\mathbb{E}[\nabla^\nu\nabla^\lambda l]$$

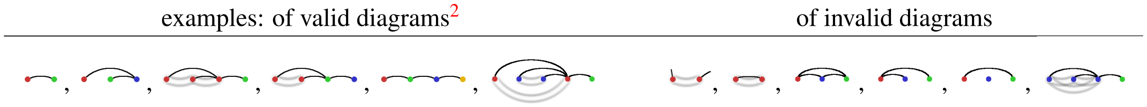
$$=-(GG+C)_{\mu\nu}J^{\mu\nu}_{\lambda}G^\lambda \qquad\qquad =-(GG+C)_{\mu\nu}H^{\mu}_{\lambda}H^{\nu\lambda}$$

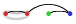
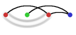
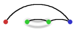
$\leftrightsquigarrow -\mathrm{uvalue}($  $)$ $\qquad$ $\leftrightsquigarrow -\mathrm{uvalue}($  $)$

[1] The η expansion's domain of convergence depends on T, so we do not expect $\lim_{\eta\to 0}$ and $\lim_{T\to\infty}$ to commute.

We notated the terms above using diagrams. Diagrams appear as we organize (1)'s terms. Each diagram evaluates to a tensor: its *un-resummed value* or ***uvalue***, defined as a product containing a dth derivative of l_x for each degree-d node, grouped under expectation symbols per the diagram's gray outlines, and tensor-contracted per the diagram's black edges (§A.4 provides details).[1] In fact, (1) is a weighted sum of the uvalues of all diagrams formally defined below.

Definition 1 (§B.3) A **diagram** is a rooted tree equipped with an equivalence relation on (i.e. a partition of) its non-root nodes. We orient the tree left-to-right so that children precede parents; the root is thus rightmost. We draw the partition structure with minimally many gray outlined ties. ◇



Since a diagram is just a rooted tree and partition,  =  =  are the same diagrams.

There are dozens of small diagrams. In many analyses, only a few diagrams are relevant. Examples: for fixed T, **to order η^d we may neglect all diagrams with more than d edges**; if $E = B = 1$, each diagram with an ancestor-descendant pair in the same part contributes zero to $\mathbb{E}[l(\theta_T)]$; for θ_0 a minimum of l, all diagrams vanish that contain a degree-1 node participating in no gray outline.[3]

2.3. Diagrams overcome the challenges of using Taylor series to study SGD

Having expressed terms in (1) as uvalues of diagrams, we seek the coefficient for each uvalue. Intuitively, a diagram represents a process (as in Figure 1) and a diagram's contribution scales with the number of ways that process may occur. Specifically, we count *embeddings*, defined below with respect to given SGD hyperparameters N, E, B.[4] We then re-state (1) as a sum over embeddings.

Definition 2 An **embedding** of a diagram is an assignment of non-root nodes to *(n,t)* pairs such that: the nth training point participates in the tth batch; parents' ts strictly exceed their children's ts; and any two nodes' ns are equal if and only if the nodes are in the same part of the partition. ◇

Key Lemma (restated) *For all T: for η sufficiently small, the final testing loss is:*

$$\mathbb{E}[l(\theta_T)] = \sum_{\substack{D \text{ a} \\ \text{diagram}}} \; \sum_{\substack{f \text{ an embed-} \\ \text{-ding of D}}} \frac{1}{\left|\mathrm{Aut}_f(D)\right|} \frac{\mathrm{uvalue}(D)}{(-B)^{|\mathrm{edges}(D)|}} \tag{2}$$

Here, $\left|\mathrm{Aut}_f(D)\right|$ counts the graph automorphisms of D that preserve f. (Typically $\left|\mathrm{Aut}_f(D)\right| = 1$.)

For example, has just one non-root node, it has as many embeddings as there are (n, t) cells where n participates in the tth update. So it has $B \cdot T$ embeddings. Since is the only diagram with one edge, it gives the full η^1 contribution to final testing loss. We thus obtain Proposition 0:

$$-(\text{\# of embeddings of } \; \text{ } \;) \cdot \mathrm{uvalue}(\text{ })/B = -T \cdot G_\mu G^\mu$$

Diagrams streamline analysis of SGD because it is in practice straightforward to count a diagram's embeddings. Moreover, the topology of diagrams has dynamical significance: $t^d T^{-p}$-th

[1] We write ⇝ instead of = since diagrams evaluate to products of cumulants C rather than of moments $GG + C$: §A.4.

[3] Recall that $E, T, B, M, N; G, H, J, C, S; \mathcal{D}, \mathcal{M}, l_x, \eta, \theta$ are defined in §1.1.

[4] and w.r.t. a deterministic selector of the tth batch. One may take expectations over such algorithms — §A.2.

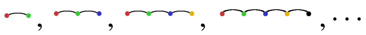
order correction[1] to the ordinary differential equation approximation of SGD is given by diagrams with d edges and p many gray outlined ties. Likewise, if we seek to isolate the effect, say, of C or H or S or J, we may consider only those diagrams that contain the corresponding subgraph in Figure **??**.

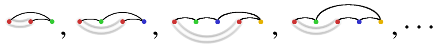
[1] We compare ODE integrated to time t to T steps of SGD with $\eta = \eta_\star t/T$ and $E = B = 1$, and we assume $p \neq 0$.

2.3.1. RESUMMATION

So far, diagrams have been a convenient but dispensible book-keeping tool. And so far, §2.1's polynomial divergence remains in (3). We now show that diagrams enable us to correct this divergence.

Let us collect similar diagrams, where our notion of "similar" permits chains to grow or shrink (see Definition 3). We obtain lists (each conveniently represented by its smallest member) such as:





We will express in closed form the total contribution to (1) of all diagrams in such a list. The idea is that the uvalues of chains are powers of hessians — e.g. uvalue( ) = GH^3G — so we may sum over chain lengths via geometric series.

Definition 3 A **link** is a degree-2 non-root node that participates in no gray outlines. E.g. has one link (green). To *reduce* at a link, we replace the link by a black edge connecting the link's two neighbors. Reduction generates an equivalence relation on diagrams. Each equivalence class has exactly one **linkless** diagram. ◇

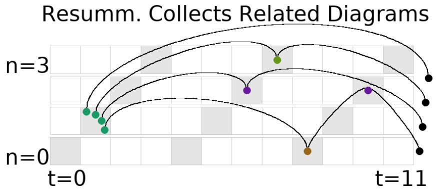


Figure 3: **Resummation propagates information damped by curvature**. Each resummed value (here, for) represents many un-resummed values, four shown here, each modulated by the Hessian ( ) in a different way.

More precisely, we define we define the *resummed value* or ***rvalue*** of an embedded diagram just as we defined the uvalue except that pairs of tensor indices are not contracted using $\eta^{\mu\nu}$. Instead, whenever an embedded edge spans Δt many time steps, we contract the two corresponding tensor indices using $(I-\eta H)^{\Delta t-1}\eta$. (For example, take Figure 3's embedding of (topmost of four). The associated uvalue is $G_\mu \eta^{\mu\nu} G_\nu$: a G for each degree-one node and an η for each edge. By contrast, the associated rvalue is $G_\mu((I-\eta H)^{12-1}\eta)^{\mu\nu}G_\nu$ since the edge spans 12 timesteps. We see that this rvalue includes uvalues for embeddings of , etc.)

2.4. Main result

Theorem 1 expresses SGD's testing loss as a sum over diagrams. A diagram with d edges scales as $O(\eta^d)$, so the following is a series in η. In practice, we truncate the series to small d (invoking Theorem 2 when possible), thus focusing on few-edged diagrams. Here we state a special case:

Theorem 1 *For any T: for η small enough, the final testing loss is a sum over* linkless *diagrams:*

$$\mathbb{E}[l(\theta_T)] = \sum_{\substack{D \text{ a linkless} \\ \text{diagram}}} \; \sum_{\substack{f \text{ an embed-} \\ \text{-ding of } D}} \frac{1}{\left|\mathrm{Aut}_f(D)\right|} \frac{\mathrm{rvalue}_f(D)}{(-B)^{|\mathrm{edges}(D)|}}$$

Remark 1 In practice, we replace sums over embeddings by integrals over t and $(I-\eta H)^t$ by $\exp(-\eta H t)$, reducing to a routine integration of exponentials at the cost of an error factor $1+o(\eta)$.◇

Theorem 2 *If $\theta_\star$is a non-degenerate local minimum of l (i.e. $G(\theta_\star) = 0$ and $H(\theta_\star) > 0$), then for SGD initialized sufficiently close to $\theta_\star$, the dth-order truncation of Theorem 1 converges as $T \to \infty$.*

Caution: the $T \to \infty$ limit in Theorem 2 might not measure any well-defined limit of SGD, since the limit might not commute with the infinite sum. We see no such pathologies in practice, so we will freely speak of "SGD in the large-T limit" as informal shorthand when referencing this Theorem.

3. Consequences of the theory

By Cor.s 1 and 2, gradient noise repels SGD. By Cor. 3, SGD senses changes in H more than SDE; in fact, (Cor. 4) SGD seeks small-H weights. Cor. 5 relates C and H to overfitting. These results do not exhaust our theory's scope; §B.7 sketches extensions to Hessian methods and natural GD.

3.1. Gradient noise repels SGD

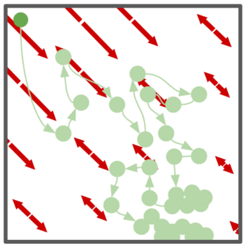


Figure 4: Chladni drift on $\mathcal{M} = \mathbb{R}^2$. Red bi-arrows depict $C(\theta)$'s major axis. SGD updates (green) tend toward small C.

Physical intuition suggests that noise repels SGD: if two neighboring regions of weight space have high and low levels of gradient noise, respectively, then the rate at which θ jumps from the former to the latter exceedsthe opposite rate. There is thus a net movement toward regions of small C.[1] Our theory makes this precise; θ drifts in the direction $-\nabla C$, and the effect is weaker when gradient noise is averaged out by large batch sizes:

Corollary 1 (Computed from **)** *SGD with $E = B = 1$ avoids high-C regions more than GD: $\mathbb{E}\left[\theta_{GD} - \theta_{SGD}\right]^{\mu} = T \cdot \frac{N-1}{4N} \nabla^{\mu} C^{\nu}_{\nu} + o(\eta^2)$.*

Roberts (2019) obtained a version of this Corollary with a nearly equal error of $O(\eta^2/N) \vee o(\eta^2)$. The Corollary's proof implies that if $\hat{l_c}$ is a smooth unbiased estimator of $\frac{N-1}{4N} C^{\nu}_{\nu}$, then GD on $l + \hat{l_c}$ has an expected testing loss that agrees with SGD's to order η^2. We call this method **GDC**.

An analogous form of averaging occurs over multiple epochs. For a tight comparison, we scale the learning rates appropriately so that, to leading order, few-epoch and many-epoch SGD agree. Then few-epoch and many-epoch SGD differ, to leading order, in their sensitivity to ∇C:

Corollary 2 (**)** *SGD with $E = B = 1$, $\eta = \eta_0$ avoids high-C regions more than SGD with $E = E_0$, $B = 1$, $\eta = \eta_0/E_0$. Precisely: $\mathbb{E}\left[\theta_{E=E_0} - \theta_{E=1}\right]^{\mu} = \left(\frac{E_0 - 1}{4E_0}\right) N \left(\nabla^{\mu} C^{\nu}_{\nu}\right) + o(\eta^2)$.*

In sum, high-C regions repel small-(E, B) SGD more than large-(E, B) SGD. We thus extend the $T = 2$ result of Roberts (2018) and resolve some questions posed therein.

3.2. SGD and SDE respond differently to changes in curvature

Ordinary and stochastic differential equations (ODE and SDE: see Liao et al. (2018)) are a popular models of SGD, but they differ from SGD in several ways. For instance, the inter-epoch noise correlations in multi-epoch SGD measurably affect SGD's final testing loss (Corollary 2), but SDE assumes uncorrelated gradient updates. Even if we restrict to single-epoch SGD, time discretization and non-Gaussian noise lead SGD and SDE to respond differently to changes in curvature. The following treats SGD with $E = B = 1$.

Corollary 3 (**,** **)** *For fixed T, SGD's final testing loss exceeds both ODE's and SDE's by $\frac{T}{2} C_{\mu\nu} H^{\mu\nu} + o(\eta^2)$. The skewness of gradient noise contributes (we work in an eigenbasis of ηH):*

$$-\frac{\eta^3}{3!} \sum_{\mu\nu\lambda} S_{\mu\nu\lambda} \frac{1 - \exp(-T\eta(H_{\mu\mu} + H_{\nu\nu} + H_{\lambda\lambda}))}{\eta(H_{\mu\mu} + H_{\nu\nu} + H_{\lambda\lambda})} J_{\mu\nu\lambda} + o(\eta^3)$$

to the excess final testing loss over SDE. This expression specializes to Proposition 1.

[1] This is the same mechanism by which sand on a vibrating plate accumulates in quiet regions (Chladni, 1787). We thus dub the SGD phenomenon the Chladni drift.

3.3. SGD descends on a landscape smoothed by the current C

Corollary 4 (Computed from **)** *Run SGD for $T \gg 1/\eta H$ from a non-degenerate test minimum. Written in an eigenbasis of ηH, θ has an expected displacement of*

$$-\frac{\eta^3}{2}\sum_{\mu\nu} C_{\mu\nu}\frac{1}{\eta(H_{\mu\mu}+H_{\nu\nu})}J_{\mu\nu\lambda}\frac{1}{H_{\lambda\lambda}} + o(\eta^2)$$

The displacement scales as $-C\nabla H$. That is, SGD moves toward minima that are flat *with respect to* C (Figure 5 ■⊞). Taking limits to drop the non-degeneracy hypothesis, we expect *sustained* motion toward flat regions in a valley of minima. In avoiding Wei and Schwab (2019)'s constant-C assumption, we find that SGD's velocity field is typically has curl (§4.2). For $\nabla(CH)$ is a total derivative but $C\nabla H$ is not. By low-pass filter theory, l increases by $CH/2 + o(C)$ when we convolve l with a C-shaped Gaussian. So we say that SGD descends on a C-smoothed landscape that changes as C does. Our $T \gg 1$ result is $\Theta(\eta^2)$, while Yaida (2019b)'s similar $T = 2$ result is $\Theta(\eta^3)$, since our analysis integrates the noise over time, hence amplifying C's effect. Experiments verify our law.

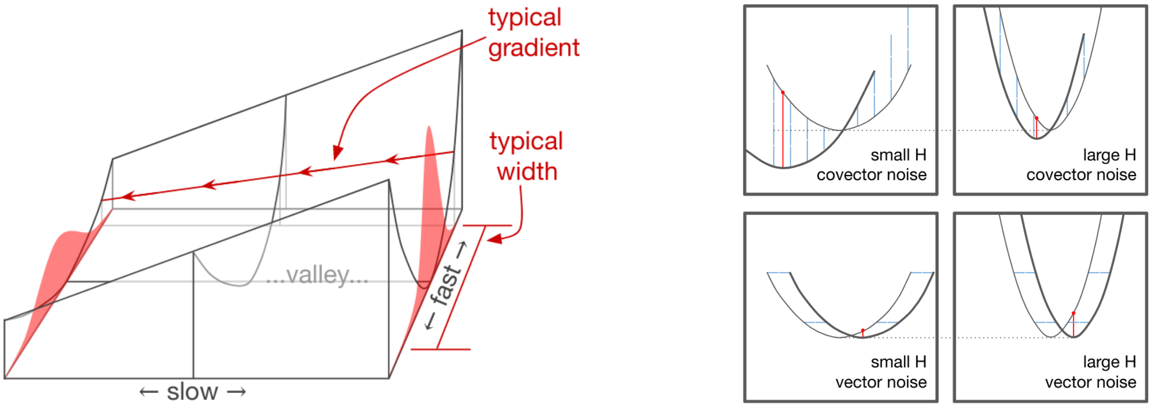


Figure 5: **Geometric intuition for curvature-noise interactions. Left**: Gradient noise pushes SGD toward flat minima (Corollary 4). Gray lines sketch a two-dimensional loss landscape near a valley of minima. Red densities show the typical θs, perturbed by noise (C) from the minimum, in two cross sections of the loss valley. $J = \nabla H$ measures curvature's change across the valley. Our theory does not assume separation between "fast" and "slow" modes, but we label them here to ease comparison with Wei and Schwab (2019). Compare with Figure 7. **Right**: Both curvature and noise structure affect overfitting. In each subplot, the ↔ axis represents weight space and the ↕ axis represents loss. Noise (blue) transforms the testing loss (thin curve) into the observed loss (thick curve). Red dots mark the testing loss at the arg-min of the observed loss. □▀: *covector*-perturbed landscapes favor large Hs. □▄: *vector*-perturbed landscapes favor small Hs. SGD's implicit regularization interpolates between these rows (Corollary 5).

3.4. Both flat and sharp minima overfit less

Intuitively, sharp minima are robust to slight changes in the average *gradient* and flat minima are robust to slight *displacements* in weight space (Figure 5 □■). However, as SGD by definition equates displacements with gradients, it may be unclear how to reason about overfitting in the presence of curvature. Our theory, by accounting for the implicit regularization of fixed-T descent, shows that both effects play a role. In fact, by routine calculus on the left hand side of Corollary 5, overfitting is maximized for medium minima with curvature $H \sim (\eta T)^{-1}$.

Corollary 5 (from ⌒, ⌒) *Initialize GD at a non-degenerate test minimum $\theta_\star$. The overfitting (testing loss minus $l(\theta_\star)$) and generalization gap (testing minus training loss) due to training are:*

$$\left(\frac{C/N}{2H}\right)^{\rho\lambda}_{\mu\nu} \left((I - \exp(-\eta T H))^{\otimes 2}\right)^{\mu\nu}_{\rho\lambda} + o(\eta^2) \quad ; \quad \left(\frac{C/N}{H}\right)^{\mu\lambda}_{\mu\nu} (I - \exp(-\eta T H))^{\nu}_{\lambda} + o(\eta)$$

The generalization gap tends to $C_{\mu\nu}(H^{-1})^{\mu\nu}/N$ as $T \to \infty$. For maximum likelihood (ML) estimation in well-specified models near the "true" minimum, $C = H$ is the Fisher metric, so we recover the AIC: (model dimension)$/N$. Unlike AIC, our more general expression is descendably smooth, may be used with MAP or ELBO tasks instead of just ML, and does not assume a well-specified model.

4. Experiments

Despite the convergence results in Theorems 1 and 2, we have no theoretical bounds for the domain and rate of convergence. Instead, we test our predictions by experiment. We perceive support for our theory in drastic rejections of the null hypothesis. For instance, in Figure 6 ▦, Chaudhari and Soatto (2018) predict a velocity of 0 while we predict a velocity of $\eta^2/6$. Here, I bars, + signs, and shaded regions all mark 95% confidence intervals based on the standard error of the mean. §C describes neural architectures, artificial landscapes, sample sizes, and further plots.

4.1. Training time, epochs, and batch size; C repels SGD more than GD

We test Theorem 1's third-order truncation on smooth convnets for CIFAR-10 and Fashion-MNIST. Theory agrees with experiment through timescales long enough for accuracy to increase by 0.5% (Figure 6 ▦, ▦). §C.7 supports Corollary 2's predictions about epoch number. Figure 6 ▦ tests Corollary 1's claim that, relative to GD, high-C regions *repel* SGD. This is significant because C controls the rate at which the generalization gap (testing minus training loss) grows (Corollary 5, Figure 6 ▦).

4.2. Minima that are flat *with respect to C* attract SGD

To test Corollary 4's C-dependence, §C.1 constructs a landscape, HELIX, on whose valley of global minima C varies. Figure 7 depicts HELIX.[1] As in Rock-Paper-Scissors, each point θ has a neighbor that is more attractive (flatter) with respect to $C(\theta)$. This permits eternal motion into the page despite the landscape's discrete translation symmetry in that direction. In §C.1, we wrap HELIX in a loop to make SGD perpetually circulate and thus to witness a velocity field with non-zero curl.

More precisely, Corollary 4 predicts a velocity of $+\eta^2/6$ per timestep, while Chaudhari and Soatto (2018)'s SDE-based analysis predicts a constant velocity of 0.[2] Our prediction agrees with experiment (Figure 6 ▦). One hopes for an "effective loss" $\tilde{l}$ such that, up to $\sqrt{T}$ diffusion terms, SGD on l mimics ODE on $\tilde{l}$. The non-zero curl in SGD's velocity field shows that no such $\tilde{l}$ exists.

[1] Thanks to Paul Seeburger's online applet, CalcPlot3D.

[2] Indeed, HELIX' velocity is η-perpendicular to the image of $(\eta C)^{\mu}_{\nu}$ in tangent space.

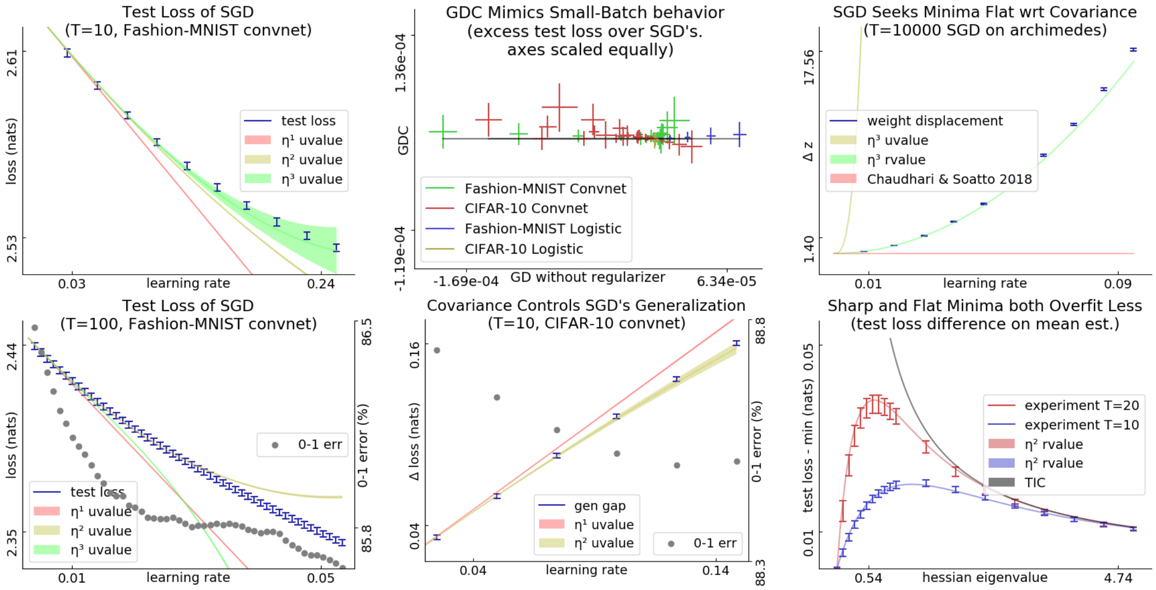


Figure 6: **Experiments on natural and artificial landscapes.** `rvalue` refers to Theorem 1's predictions, approximated as in Remark 1. `uvalues` are simpler but (see □□■/□□□) less accurate.
Left: Perturbation models SGD for small ηT. Fashion-MNIST convnet's testing loss vs learning rate. In this small T setting, we choose to use our theory's simpler un-resummed values (A.4) instead of the more precise rvalues. ■□□/□□□: For all initializations tested (1 shown, 11 unshown), the order 3 prediction agrees with experiment through $\eta T \approx 10^0$, corresponding to a decrease in 0-1 error of $\approx 10^{-3}$. □□□/■□□: For large ηT, our predictions break down. Here, the order 3 prediction holds until the 0-1 error improves by $5 \cdot 10^{-3}$. Beyond this, 2nd order agreement with experiment is coincidental.
Center: C controls generalization gap. With equal-scaled axes, □■□/□□□ shows that GDC matches SGD (small vertical variance) better than GD matches SGD (large horizontal variance) in testing loss for a range of η ($\approx 10^{-3} - 10^{-1}$) and initializations (zero and several Xavier-Glorot trials) for logistic regression and convnets. Here, $T = 10$. □□□/□■□: CIFAR-10 generalization gaps. For all initializations tested (1 shown, 11 unshown), the degree-2 prediction agrees with experiment through $\eta T \approx 5{\cdot}10^{-1}$.
Right: Predictions near minima excel for large ηT. □□■/□□□: SGD traverses HELIX' valley of global minima. Note: H and C are bounded across the valley, we see drift for all small η, and we see displacement exceeding the landscape's period of 2π. So: the drift is not a pathology of well-chosen η, of divergent noise, or of ephemeral initial conditions. □□□/□□■: For MEAN ESTIMATION with fixed C and a range of Hs, initialized at the truth, the testing losses after fixed-T GD are smallest for very sharp and very flat H. Near $H = 0$, our predictions improve on AIC,TIC (Dixon and Ward, 2018).

4.3. Sharp and flat minima both overfit less than medium minima

Prior work (§5.1) finds both that SGD leads to overfits less near *sharp* minima (for, l^2 regularization sharpens minima) or that SGD overfits less near *flat* minima (for, flat minima are robust to small displacements). In fact, both phenomena occur, and noise structure determines which dominates (Corollary 5). This effect appears even in MEAN ESTIMATION (§C.1): Figure 6 □□□/□□■. §C.7 also presents preliminary results suggesting that Corollary 5's generalization gap estimate may serve as a regularizer term enabling descent-based tuning of hyperparameters such as a model's l_2 coefficients.

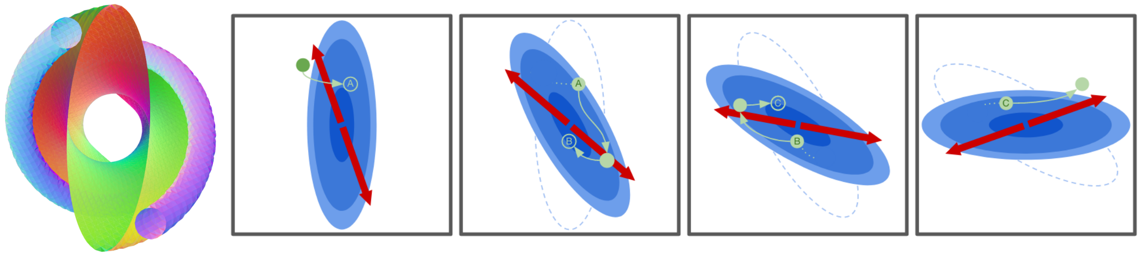
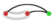
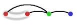
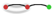


Figure 7: **Leftmost**: The HELIX landscape is defined on a three-dimensional weight space that extends indefinitely into and out of the page. A helical level surface (orange-green) of l winds around its axis, a one-dimensional valley of minima perpendicular to the page. l is large outside this surface. Gradient noise is parallel to the page and tends to point from the valley toward the outer two tubes. **Rightmost four**: HELIX induces SGD to move into the page. Green dots trace a trajectory over four cross sections of weight space that descend progressively into the page. In blue are partial contour maps of l; the valley of minima intersects each pane's dark blue center. Dotted blue curves help to compare adjacent panes. Red arrows show the major axis of gradient noise in each pane. **Green trajectory, explained**: Pane ■□□□ shows θ, initialized at the dark green dot, following l's gradient toward point A in the next pane (deeper into the page). Next (□■□□), gradient noise kicks θ from point A, leading θ to fall toward point B. Such falling continues in □□■□, even though the gradient noise happens to oppose the previous pane's. □□□■ shows θ at point C kicked uphill by gradient noise; θ never settles and the phenomena depicted here continue for all time.

5. Conclusion

This paper presents a new physics-inspired perspective on SGD. We use diagrams to study stochastic optimization on short timescales or near minima. Corollaries 4 and 5 together show that SGD avoids curvature and noise, which to leading order control generalization.

Analyzing , we proved that **flat and sharp minima both overfit less** than medium minima. Intuitively, flat minima are robust to vector noise, sharp minima are robust to covector noise, and medium minima robust to neither. We thus proposed a regularizer enabling gradient-based hyperparameter tuning. Inspecting , we extended Wei and Schwab (2019) to nonconstant, non-isotropic covariance to reveal that **SGD descends on a landscape smoothed by the current covariance** C. As C evolves, the smoothed landscape evolves, resulting in non-conservative dynamics. Examining , we showed that **GD may emulate SGD**, as suggested by Roberts (2018). This 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).

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. Thus, the theory might aid future analysis of fine-tuners such as Finn et al. (2017)'s MAML.

Much as meteorologists understand how warm and cold fronts interact despite long-term forecasting's intractability, we quantify how curvature and noise contribute to counter-intuitive dynamics governing each short-term interval of SGD's trajectory. Equipped with our theory, users of deep learning may refine intuitions — e.g. that SGD descends on the training loss — to account for noise.

5.1. Related work

Kiefer and Wolfowitz (1952) united gradient descent (Cauchy, 1847) with stochastic approximation (Robbins and Monro, 1951) to invent SGD. Since the development of back-propagation for efficient differentiation (Werbos, 1974), SGD has been used to train connectionist models, e.g. neural networks (Bottou, 1991), recently to remarkable success (LeCun et al., 2015).

Several lines of work treat the overfitting of SGD-trained networks (Neyshabur et al., 2017a). For example, Bartlett et al. (2017) controls the Rademacher complexity of deep hypothesis classes, leading to optimizer-agnostic generalization bounds. Yet SGD-trained networks generalize despite their ability to shatter large sets (Zhang et al., 2017), so generalization must arise from not only architecture but also optimization (Neyshabur et al., 2017b).

Some analyses of optimization's implicit regularization use a Langevin dynamics or SDE approximation (e.g. Chaudhari and Soatto (2018); Zhu et al. (2019)), but, per Yaida (2019a), such continuous-time or uncorrelated-noise analyses treat SGD noise incorrectly. We avoid these pitfalls by Taylor expanding around $\eta = 0$ as in Roberts (2018). Unlike that work, we generalize beyond order η^1 and $T = 2$. To do so, we develop new summation techniques with improved large-T convergence. Our interpretion of the resulting terms offers a new qualitative picture of SGD as a superposition of simpler information-flow processes.

Other research focuses on *double descent* and suggests that some highly overparameterized models share implicit regularization properties with linear least-squares models (Belkin et al., 2019), for example by bounding log-determinants (and hence the effective dimensions) of feature matrices and weight spaces (Mei and Montanari, 2020).[1] Our work reveals new dynamics toward and within valleys of minima, dynamics that may also reduce the effective dimension of model space. However, our focus on the structure of gradient noise may be overspecific, since recent work finds that GD and SGD may both converge to the same set of global minima (Zou et al., 2020) or that noise covariance but not higher moments are relevant to regularization (Wu et al., 2020).

Our predictions are vacuous for large η. Other analyses treat large-η learning phenomenologically, whether by finding empirical correlates of the generalization gap (Liao et al., 2018), by showing that *flat* minima generalize (Hoffer et al. (2017), Keskar et al. (2017), Wang et al. (2018), Zhu et al. (2019)), or by showing that *sharp* minima generalize (Stein (1956), Dinh et al. (2017), Wu et al. (2018),). We find that SGD's implicit regularization mediates between these seemingly clashing intuitions (§4.3).

Prior work analyzes SGD perturbatively: Dyer and Gur-Ari (2019) perturb in inverse network width, using 't Hooft diagrams to correct the Gaussian Process approximation for specific nets. Perturbing to order η^2, Chaudhari and Soatto (2018) and Li et al. (2017) assume uncorrelated Gaussian noise. By contrast, we use Penrose diagrams Penrose (1971) to compute testing losses to arbitrary order in η. We allow correlated, non-Gaussian noise and thus *any* smooth architecture. E.g. we assume no information-geometric relationships between C and H,[2] so we may model VAEs.

[1] Mei and Montanari (2020)'s eq. 75 bounds a log-determinant defined in eq. 61 of a transformed feature matrix. Compare to linear Representer Theorems (Mohri et al., 2018).

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

References

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

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

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

M. Belkin, Daniel Hsu, Siyuan Ma, and S. Mandal. Reconciling modern machine learning practice and the bias-variance trade-off. *PNAS*, 2019.

S. Bonnabel. Sgd on riemannian manifolds. *IEEE Transactions on Automatic Control*, 2013.

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

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

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

E.F.F. Chladni. Entdeckungen über die theorie des klanges. *Leipzig*, 1787.

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

M.F. Dixon and T. Ward. Takeuchi information as a form of regularization. *Arxiv Preprint*, 2018.

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

F. Dyson. The radiation theories of tomonaga, schwinger, and feynman. *Physical Review*, 1949.

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

C.F. Gauss. Theoria combinationis obsevationum erroribus minimis obnoxiae, section 39. *Proceedings of the Royal Society of Gottingen*, 1823.

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

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

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

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

I. Kolář, P.W. Michor, and J. Slovák. Natural operations in differential geometry. *Springer*, 1993.

A. Krizhevsky. Learning multiple layers of features from tiny images. *UToronto Thesis*, 2009.

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

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

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

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

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms i. *PMLR*, 2017.

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

Song Mei and A. Montanari. The generalization error of random features regression. *Arxiv Preprint*, 2020.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. Foundations of machine learning, section 6.3.2. *MIT Press*, 2018.

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

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

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

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

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, T. Killeen, Zeming Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, Edward Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, Lu Fang, Junjie Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

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

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

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

D.A. Roberts. Sgd. *Personal communication*, 2019.

G.-C. Rota. Theory of möbius functions. *Zeitschrift für Wahrscheinlichkeitstheoriei und Verwandte Gebiete*, 1964.

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

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

Huan Wang, N.S. Keskar, Caiming Xiong, and R. Socher. Identifying generalization properties in neural networks. *Arxiv Preprint*, 2018.

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

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

Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, V. Braverman, and Zhanxing Zhu. On the noisy gradient descent that generalizes as sgd. *ICML*, 2020.

Lei Wu, Chao Ma, and Weinan E. How sgd selects the global minima in over-parameterized learning. *NeurIPS*, 2018.

Han Xiao, L. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *Arxiv Preprint*, 2017.

Sho Yaida. Fluctuation-dissipation relations for sgd. *ICLR*, 2019a.

Sho Yaida. A first law of thermodynamics for sgd. *Personal Communication*, 2019b.

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

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

Zhanxing Zhu, Jingfeng Wu, Bing Yu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent. *ICML*, 2019.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *MLJ*, 2020.

Organization of the appendices

The following three appendices serve three respective functions:

- to explain how to calculate using diagrams;
- to prove our results (and pose a conjecture);
- to specify our experimental methods and results.

In more detail, we organize the appendices as follows.

Appendix A. Tutorial: how to use diagrams

This paper presents a new 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 are **four steps** to computing the expected testing loss, or other quantities of interest, after a specific number of gradient updates:

- **Specify, as a grid**, the batch size, training set size, and number of epochs.
- **Draw embeddings**, of diagrams into the grid, as needed for the desired precision.
- **Evaluate each diagram embedding**, whether exactly (via rvalues) or roughly (via uvalues).
- **Sum the embeddings' values** to obtain the quantity of interest as a function of η.

After presenting two example calculations that follow these four steps, we detail each step individually. Though we focus on the computation of expected testing losses, we describe how the four steps may give us other quantities of interest: variances instead of expectations, training statistics instead of testing statistics, or weight displacements instead of losses.

A.1. Two example calculations

We illustrate the four step procedure above by using it to answer the following two questions.

Our first example calculation reproduces Proposition 0. In other words, it answers the question:

Question 1 (Leading order effect of gradients) *What's the leading order loss decrease* $\mathbb{E}[l(\theta_T) - l(\theta_0)]$*? We seek an answer expressed in terms of the landscape statistics at initialization:* $G, H, C, \cdots$*. We expect only* G *to be relevant.*

Our second example is (an illustrative case of) Corollary 2.

Question 2 (Leading order effect of epochs) *How does multi-epoch SGD differ from single-epoch SGD? Specifically, what is the difference between the final testing losses of the following two versions of SGD?*

- *SGD over* $T = M_0 \times N$ *time steps, learning rate* η_0/M*, and batch size* $B = 1$
- *SGD over* $T = N$ *time steps, learning rate* η_0*, and batch size* $B = 1$

We seek an answer expressed in terms of the landscape statistics at initialization: $G, H, C, \cdots$*.*

To make our discussion concrete, we will set $M_0 = 2$; our analysis generalizes directly to larger M_0.

We scaled the above two versions of SGD deliberately, to create an interesting comparison. Specifically, on a noiseless linear landscape $l_x = l \in (\mathbb{R}^n)^*$, the versions attain equal testing losses, namely $l(\theta_0) - T l_\mu \eta^{\mu\nu}$. So Question 2's answer will be second-order (or higher-order) in η.

A.1.1. Computations: Grids

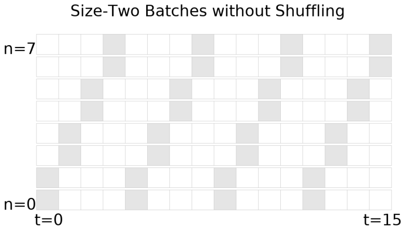
We begin by asking a question about the final testing loss of some form of SGD. We specify the batch size, training set size, and number of epochs of the setting under analysis by drawing an appropriate grid. That is, we

- **draw an $N \times T$ grid** and
- shade its cells, shading the (n, t)th cell **when the tth batch includes the nth data point**.

Thus, each column contains B (batch size) many shaded cells and each row contains E (epoch number) many shaded cells.

Effect of Gradients (Question 1)

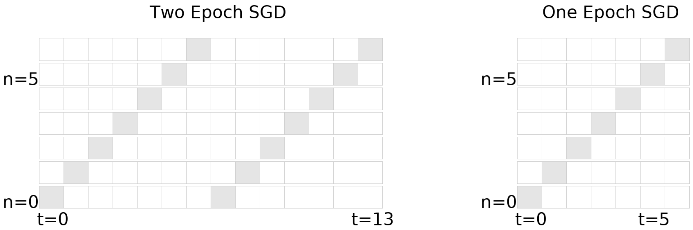
Question 1 does not specify a batch size, epoch number, or training set size and so does not specify a grid. In fact, we wish to answer the Question for any choice of those hyperparameters. E.g. we'll answer the Question for SGD with hyperparameters $B, E, N = 2, 4, 8$:



A grid for SGD with batch size $B = 2$ run for $E = 4$ epochs on $N = 8$ training points for a total of $T = 16$ timesteps.

Effect of Epochs (Question 2)

Two grids are relevant to Question 2: one for multi-epoch sgd and another for single-epoch SGD. See below.



Grids for single-epoch and multi-epoch SGD. Both grids depict $N = 7$ training points and batch size $B = 1$. **Left**: SGD with $M = 2$ update per training sample for a total of $T = MN = 2N$ many updates. **Right**: SGD with $M = 1$ update per training sample for a total of $T = MN = N$ many updates.

A.1.2. Computations: Embeddings of diagrams into grids

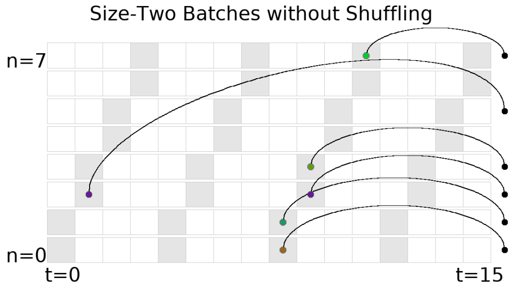
Say we permit order-d errors. We draw all relevant diagrams with d or fewer edges and then characterize the embeddings of those diagrams in §A.1.1's grid. An *embedding* of a diagram D in a grid is an assignment of D's non-root nodes to shaded cells (n, t) obeying the following criteria:

- **time-ordering condition**: the times t strictly increase along each path from leaf to root; and
- **correlation condition**: if two nodes are in the same part of D's partition, then they are assigned to the same datapoint n.

We draw embeddings by placing nodes in their assigned shaded (n, t)cells; we draw the root nodes outside the grids (at arbitrary positions).

Effect of Gradients (Question 1)

We seek an order 1 result and thus consider one-edged diagrams; there is only one: . We now describe the embeddings of this diagram:

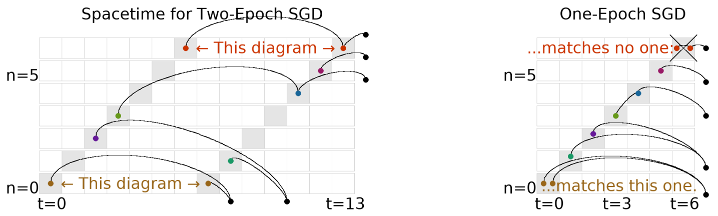


has one embedding for each shaded cell. An embedding of a non-root node at cell (n, t) represents the influence of the datapoint n on the testing loss due to the tth update. To first order (i.e. one-edged diagrams), the influences of different timesteps do not interact. The combinatorics of embeddings is thus straightforward.

Effect of Epochs (Question 2)

We seek an order 2 result and thus consider two-edged diagrams; there are four: , , , and . The figure below shows some embeddings of order-1 and order-2 diagrams (i.e. one-edged and two-edged diagrams) into the grid relevant to Question 2. Specifically, from top to bottom in each grid, the five diagrams embedded are (or), , , , and (or).

The diagram may be embedded wherever the diagram may be embedded, but not vice versa. Likewise for and .



Here, **embeds into the multi-epoch but not single-epoch grid. Left**: embeds into the multi-epoch grid. **Right**: cannot embed into the single-epoch grid: the correlation condition forces both red nodes into the same row and thus the same cell; the time-ordering condition forces the red nodes into distinct columns and thus distinct cells.

A.1.3. Computations: Evaluating each diagram embedding

We evaluate §A.1.2's diagrams. We choose here to compute uvalues (apt for fixed T), not rvalues. These rules translate diagrams to numbers:

- **Node rule**: Replace each degree d node by $\nabla^d l_x$.
- **Outline rule**: surround the nodes in each part of the partition by a "cumulant bracket". If a part contains one node x, the cumulant bracket is the expectation: $\mathbb{E}[x]$. If the part contains two nodes x, y, the cumulant bracket is the covariance: $\mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]$.[1]
- **Edge rule**: insert a $\eta^{\mu\nu}$ for each edge. The indices μ, ν should match the corresponding indices of the two nodes incident to the edge.

Effect of Gradients (Question 1)

In §A.1.2 we determined the embeddings of . Now we evaluate uvalue( ). The node rule suggests that we begin with

$$\nabla_\mu l_x \nabla_\nu l_x$$

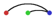
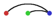
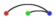
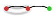
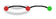
We have two factors, each with one derivative because the diagram has two nodes, each of degree one. Note that the number of indices (here, two) is the total degree over all nodes and thus also twice the number (here, one) of edges. The outline rule transforms this to

$$\mathbb{E}\left[\nabla_\mu l_x\right]\mathbb{E}\left[\nabla_\nu l_x\right] = G_\mu G_\nu$$

since all parts in the diagram's partition have size one. The edge rule inserts a factor $\eta^{\mu\nu}$ to yield:

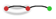
uvalue( ) $= G_\mu G_\nu \eta^{\mu\nu} = G_\mu G^\mu$

Effect of Epochs (Question 2)

In §A.1.2 we saw that  embeds similarly into multi-epoch and single-epoch grids: its multi-epoch embeddings correspond by a $M_0^2 : 1$ map to its single-epoch embeddings. Since we scaled the learning rate of the two SGD versions by a factor of M_0, and since  (being two-edged) scales as η^2, *the total* uvalue *of its multi-epoch embeddings will match the total* uvalue *of its single-epoch embeddings.* So we need not compute 's contribution.
We see that this cancellation happens for all of the order-2 diagrams *except* for . Therefore, we must only compute uvalue().
The node rule suggests that we begin with $\nabla_\mu l_x \nabla_\nu \nabla_\lambda l_x \nabla_\rho l_x$. The outline rule transforms this to

$$\left(\mathbb{E}\left[\nabla_\mu l_x \nabla_\nu \nabla_\lambda l_x\right] - \mathbb{E}\left[\nabla_\mu l_x\right]\mathbb{E}\left[\nabla_\nu \nabla_\lambda l_x\right]\right)\mathbb{E}\left[\nabla_\rho l_x\right] = (\nabla_\nu C_{\mu\lambda}/2) G_\rho$$

The edge rule inserts a factor $\eta^{\mu\lambda}\eta^{\nu\rho}$ to yield:

uvalue() $= (\nabla_\nu C_{\mu\lambda}/2) G_\rho \eta^{\mu\lambda}\eta^{\nu\rho} = G^\nu \nabla_\nu C^\mu_\mu / 2$

[1] The general pattern is that the cumulant bracket $\mathbb{C}[\prod_{i\in I} x_i]$ of a product indexed by I is (here, P ranges over partitions of I with at least two parts; $I = \sqcup_{p\in P} p$):

$$\mathbb{C}[\prod_{i\in I} x_i] = \mathbb{E}[\prod x_i] - \sum_{\text{partition } P} \prod_{p\in P} \mathbb{C}[\prod_{i\in p} x_i]$$

A.1.4. Computations: Summing the embeddings' values

Our Key Lemma('s restatement) says that to compute a testing loss,

- we sum §A.1.3's uvalues, each weighted by the number of ways its diagram embeds in the grid,
- where embeddings with s many symmetries count only $1/s$ much toward the total number of embeddings.

A symmetry of an embedding f of a diagram D, i.e. an element of $\text{Aut}_f(D)$, is defined to be a relabeling of D's nodes that simultaneously preserves D's rooted tree structure, D's partition structure, and f's assignment of nodes to (n, t) cells of the grid. This is a strong constraint, so there will typically be no symmetries except for the identity, meaning that $s = 1$.

Effect of Gradients (Question 1)

Referring again to §A.1.2, we see that $D =$ has TB many embeddings (B many embeddings per column for T many columns). Since D has no non-trivial automorphisms (i.e. no non-trivial relabeling of nodes that preserves the root, the graph structure, and the equivalence relation on non-root nodes), D has no non-trivial automorphisms that preserve any given embedding. Thus $|\text{Aut}_f(D)| = 1$ for each embedding of D. We conclude that the Restated Key Lemma's expression (3)

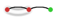
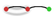
$$\sum_{\substack{D \text{ a}\\ \text{diagram}}} \sum_{\substack{f \text{ an embed-}\\ \text{-ding of } D}} \frac{(-B)^{-|\text{edges}(D)|}}{\left|\text{Aut}_f(D)\right|} \text{uvalue}(D)$$

has as its contribution from $D =$ the value

$$(\#\text{of embeddings } f) \cdot \frac{(-B)^{-1}}{1} G_\mu G^\mu = TB \cdot (-G_\mu G^\mu / B)$$

Proposition 0's expression $-TG_\mu G^\mu$ follows.

Effect of Epochs (Question 2)

Referring again to §A.1.2, we see that  has $\binom{M_0}{2} N$ many embeddings into the multi-epoch grid (one embedding per pair of distinct epochs, per row) — and no embeddings into the single-epoch grid. Moreover, each embedding of  has $\left|\text{Aut}_f(D)\right| = 1$. We conclude that the testing loss of $M = M_0$ SGD exceeds the testing loss of $M = 1$ SGD by this much:

$$\binom{M_0}{2} N \cdot \frac{(-1)^2}{1} \cdot (\nabla_\nu C_{\mu\lambda}/2) G^\rho \eta^{\mu\lambda} \eta^{\nu\rho} + o(\eta^2)$$

Since Question 2 defines $\eta^2 = \eta_0^2 / M_0^2$, we can rewrite our answer as:

$$l(\theta_{M=M_0, \eta=\eta_0/M_0}) - l(\theta_{M=1, \eta=\eta_0}) = \frac{M_0 - 1}{4M_0} N \cdot G^\nu (\nabla_\nu C^\mu_\mu) + o(\eta_0^2)$$

where we use η_0 to raise indices. This completes the example problem.

A.2. How to identify the relevant grid

Diagrams tell us about the loss landscape but not about SGD's batch size, number of epochs, and training set size We encode this SGD data as a set of pairs (n,t), where we have one pair for each participation of the nth datapoint in the tth update. For instance, full-batch GD has NT many pairs, and singeleton-batch SGD has T many pairs. We will draw these (n,t) pairs as shaded cells in an $N \times T$ grid; we will call the shaded grid the SGD's **grid**. See Figure 8.

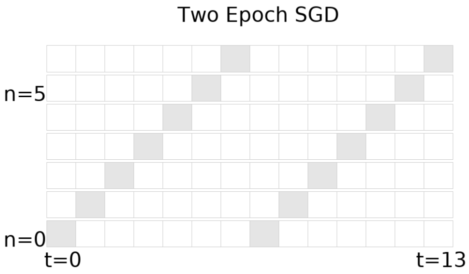
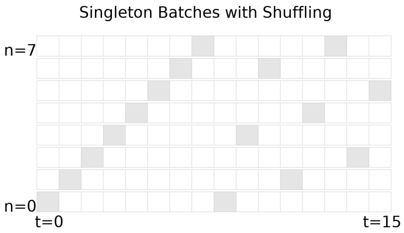


Figure 8: **The grids of two SGD variants.** Shaded cells show (n,t) pairs (see text). **Left**: Two epoch SGD with batch size one. **Right**: Four epoch SGD with batch size two.

When using the diagram method to solve a problem relating to SGD with batch size B and E many epochs (over T many time steps and on N many training samples), one shades the cells of an $N \times T$ grid with B shaded cells per column and E shaded cells per row.

Note: A grid may also depict the inter-epoch permuting of training sets due to which the bth batch in one epoch differs from the bth batch in a different epoch. For instance, see the grid to the right. Since each grid commits to a concrete sequence of training set permutations, we may analyze SGD with randomized permutations by taking expectations over multiple grids. However, the corollaries in this text are invariant to inter-epoch training set permutations, so we will not focus on this point.



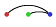
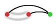
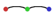
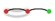
A.3. How to identify the relevant diagram embeddings

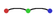
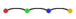
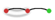
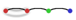
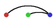
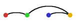
We explain below what the words mean in this green summary box.

When using the diagram method to compute SGD's final testing loss to order $o(\eta^d)$, we consider all diagrams with d or fewer edges and that have a non-zero number of embeddings into the relevant grid.

If we seek the isolated contribution due to a landscape statistic (e.g. due to $J =$), we may ignore all diagrams that contain that subgraph. If we are in a setting where a certain landscape statistic vanishes, (e.g., at a minimum: $G(\theta_0) =$ vanishes), we may neglect all diagrams that contain that subgraph.

If we are using rvalues (see next section for discussion of rvalues and uvalues), then we consider only the linkless diagrams. For each diagram, we must enumerate the embeddings, i.e. the assignments of the diagram's nodes to grid cells that obey both the time-ordering condition and correlation condition.

A *diagram* is a finite rooted tree equipped with a partition of its nodes, such that the root node occupies a part of size 1. For example, there are four diagrams with two edges: , , , and . As always, we specify a diagram's root by drawing it rightmost.

A diagram is *linkless* when each of its degree-2 nodes is in a part of size one. Intuitively, this rules out multi-edge chains unadorned by fuzzy ties. Thus, only the first diagram in the list , , , ⋯ is linkless. Only the first diagram in the list , , ⋯ is linkless. Only the first diagram in the list , , ⋯ is linkless.

An *embedding* of a diagram D into a grid is an assignment of D's non-root nodes to shaded cells (n,t) that obeys the following two criteria:

- **time-ordering condition**: the times t strictly increase along each path from leaf to root; and
- **correlation condition**: if two nodes are in the same part of D's partition, then they are assigned to the same datapoint n.

We may conveniently draw embeddings by placing nodes in the shaded cells to which they are assigned. Then, the time-ordering condition forbids (among other things) intra-cell edges, and the correlation condition demands that fuzzily tied nodes are in the same row. See Figure 9.

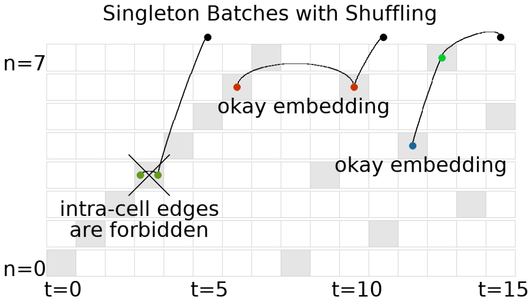
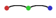
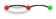
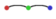
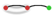
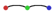
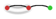
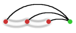
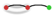


Figure 9: Embeddings, legal and illegal. **Left**: illegal embedding of , since the time-ordering condition is not obeyed. For the same reason, not a legal embedding of . **Middle**: an embedding of . Also an embedding of , since the correlation condition is obeyed. **Right**: a legal embedding of . Not an embedding of , since the correlation condition is not obeyed.

In principle, the relevant diagrams for a calculation with error $o(\eta^d)$ are the diagrams with at most d edges. For d greater than 2, there will be many such diagrams. However, in practice we gain insight even from considering one diagram at a time:

Remark *In this paper's corollaries, we seek to extract the specific effect of a specific landscape or optimization feature such as skewed noise (Proposition 1) or multiple epochs (§A.1). In these cases, it is usually the case that most diagrams are irrelevant. For example, because a diagram evaluates to a product of its components, the only way the skewness of gradient noise can appear in our calculations is through diagrams such as*  *that have a part of size* 3. *Likewise, in §A.1 we argued by considering which embeddings that the only diagram relevant to Question 2 is* . ◇

Here are some further examples. Table 1 shows the 6 diagrams that may embed into the grid of $E = B = 1$. It shows each diagram in multiple ways to underscore that diagrams are purely topological and to suggest the ways in which these diagrams may embed into a grid.

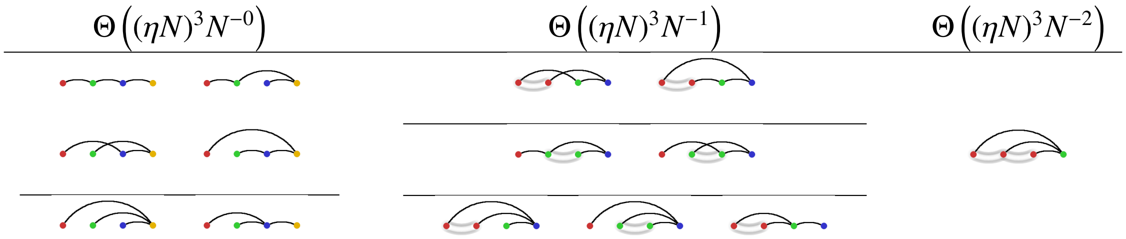

$\Theta\left((\eta N)^3 N^{-0}\right)$
$\Theta\left((\eta N)^3 N^{-1}\right)$
$\Theta\left((\eta N)^3 N^{-2}\right)$

Table 1: **Multiple ways to draw the** 6 **distinct degree-**3 **diagrams for** $B = E = 1$ **SGD's testing loss.** Because the grid of $B = E = 1$ SGD has only one cell per row and one cell per column, the only diagrams that have a non-zero number of embeddings are the diagrams such that each ancestor-descendant pair in the rooted tree occupie two different parts of the partition. We show $(4 + 2) + (2 + 2 + 3) + (1)$ ways to draw the 6 diagrams. In fact, these drawings show all of the time-orderings of the diagrams' nodes that are consistent with the time-ordering condition. **Organization**: We organize the diagrams into columns by the number of parts in their partitions. Because partitions (fuzzy outlines) indicate correlations between nodes (i.e. noise), diagrams with fuzzy outlines show deviations of SGD away from deterministic ODE. The big-Θ notation that heads the columns gives the asymptotics of the sum-over-embeddings of each diagram's uvalues (for N large and η small even relative to $1/N$). **Left**: Diagrams for ODE behavior. **Center**: 1st order deviation of SGD away from ODE. **Right**: 2nd order deviation of SGD from ODE with appearance of non-Gaussian statistics.

A.4. How to evaluate each embedding

We will discuss how to compute both rvalues and uvalues. Both are ways of turning a diagram embedding into a number. The paper body mainly mentions rvalues. uvalues are simpler to calculate, since they depend only on a diagram's topology, not on the way it is embedded. rvalues are more accurate; in particular, when we initialize near a non-degenerate local minimum, rvalues do not diverge to ±∞ as $T \to \infty$.

We will explain the following green summary box.

TURN AN EMBEDDING OF A DIAGRAM into its uvalue or rvalue by applying the following rules.
- **Node rule**: Replace each degree d node by $\nabla^d l_x$.
- **Outline rule**: surround the nodes in each part of the partition by a "cumulant bracket". If a part contains one node x, the cumulant bracket is the expectation: $\mathbb{E}[x]$. If the part contains two nodes x, y, the cumulant bracket is the covariance: $\mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]$.[1]
- If we wish to compute a uvalue, then we apply: **Edge rule for uvalues**: insert a $\eta^{\mu\nu}$ for each edge. The indices μ, ν should match the corresponding indices of the two nodes incident to the edge.
- If we wish to compute an rvalue, then we apply **Edge rule for rvalues**: if an edge's endpoints are embedded to times t, t', insert a factor of $K^{|t'-t|-1}\eta$, where $K \triangleq (I - \eta H)$. Here, we consider the root node as embedded to the time T.

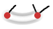
A.4.1. UN-RESUMMED VALUES: uvalue(D)

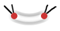
Each part in a diagram's partition looks like one of the following fragments (or one of the infinitely many analogous fragments):

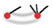
$G \triangleq \mathbb{E}_x[\nabla l_x(\theta)] \triangleq$

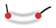
$H \triangleq \mathbb{E}_x[\nabla\nabla l_x(\theta)] \triangleq$

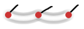
$J \triangleq \mathbb{E}_x[\nabla\nabla\nabla l_x(\theta)] \triangleq$

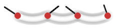
$\mathbb{E}_x[(\nabla l_x(\theta) - G)(\nabla\nabla l_x(\theta) - H)] \triangleq$ 

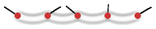
$\mathbb{E}_x[(\nabla\nabla l_x(\theta) - H)(\nabla\nabla l_x(\theta) - H)] \triangleq$ 

$\mathbb{E}_x[(\nabla l_x(\theta) - G)(\nabla\nabla\nabla l_x(\theta) - J)] \triangleq$ 

$C \triangleq \mathbb{E}_x\left[(\nabla l_x(\theta) - G)^2\right] \triangleq$ 

$S \triangleq \mathbb{E}_x\left[(\nabla l_x(\theta) - G)^3\right] \triangleq$ 

$\mathbb{E}_x\left[(\nabla l_x(\theta) - G)^4\right] - 3C^2 \triangleq$ 

$\mathbb{E}_x\left[(\nabla l_x(\theta) - G)^5\right] - 10CS \triangleq$ 

The above examples illustrate the **Node rule**: each degree d node evaluates to $\nabla^d l_x$.

Fuzzy outlines dictate how to collect the $\nabla^d l_x$s into expectation brackets. For example, we could collect the nodes within each part (of the partition) into a pair of expectation brackets $\mathbb{E}_x[\cdot]$ — call the result the **moment value**. However, this would yield (un-centered) moments such as $\mathbb{E}_x\left[(\nabla l_x(\theta))^2\right]$ instead of cumulants such as $C = \mathbb{E}_x\left[(\nabla l_x(\theta) - G)^2\right]$. For technical reasons (§B.4 and §B.5), cumulants will be easier to work with than moments, so we will choose to define the values of diagrams slightly differently as follows.[1]

Outline rule: surround the nodes in each part of the partition by a "cumulant bracket". The cumulant bracket $\mathbb{C}[\prod_{i\in I} x_i]$ of a product indexed by I is (here, P ranges over partitions of I with at least two parts; $I = \sqcup_{p\in P} p$):

$$\mathbb{C}[\prod_{i\in I} x_i] = \mathbb{E}[\prod x_i] - \sum_{\text{partition } P} \prod_{p\in P} \mathbb{C}[\prod_{i\in p} x_i]$$

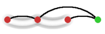
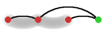
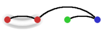
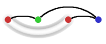
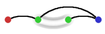
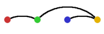
Thus, a cumulant bracket of a diagram is the moment bracket of that diagram minus other terms. Those other terms are obtained by considering diagrams with the same graph structure but strictly more parts in their partition. The recursive definition of $\mathbb{C}$ grounds out because the maximal number of parts in a partition of a finite set is finite.

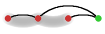
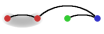
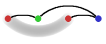
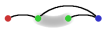
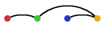
For example, if a part contains one node x, the cumulant bracket is the expectation: $\mathbb{E}[x]$. If the part contains two nodes x, y, the cumulant bracket is the covariance: $\mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]$. If a part contains three nodes x, y, z, then the cumulant bracket is

$$\mathbb{E}[xyz] - \mathbb{E}[x]\mathbb{E}[yz] - \mathbb{E}[y]\mathbb{E}[xz] - \mathbb{E}[z]\mathbb{E}[yz] + 2\mathbb{E}[x]\mathbb{E}[y]\mathbb{E}[z]$$

We visualize the above in the following example:

Example 1 For example, if we denote moment values by solid gray fuzzy ties (instead of fuzzy outlines), then:

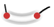
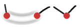
 $\triangleq$  $-$  $-$  $-$  $-$ 

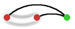
$\triangleq$  $-$  $-$  $-$  $+ 2$ 

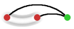
We will use the concept of "moment values" again in §B.4. ◇

[1] This is just the standard Möbius recursion for defining cumulants (see Rota (1964)).

Finally, we come to edges. **Edge rule**: insert a factor of $\eta^{\mu\nu}$ for each edge. The indices μ, ν should match the corresponding indices of the two nodes incident to the edge.

Example 2 (Un-resummed value) Remember that  $= C_{\mu\nu}$ and $= H_{\lambda\rho}$, so that  $= C_{\mu\nu}H_{\lambda\rho}$. Then

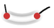
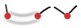
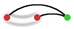
uvalue() $= C_{\mu\nu}H_{\lambda\rho}\eta^{\mu\lambda}\eta^{\nu\rho}$

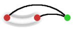
Here,  has two edges, which correspond in this example to the tensor contractions via $\eta^{\mu\lambda}$ and via $\eta^{\nu\rho}$, respectively. ◇

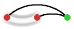
A.4.2. RESUMMED VALUES: $\text{rvalue}_f(D)$

The only difference between rvalues and uvalues is in their rule for evaluating edges.

Edge rule: if an edge's endpoints are embedded to times t, t', insert a factor of $K^{|t'-t|-1}\eta$, where $K \triangleq (I - \eta H)$. Here, we consider the root node as embedded to the time T.

Example 3 (Re-summed value) Recall as in Example 2 that  $= C_{\mu\nu}$ and $= H_{\lambda\rho}$, so that  $= C_{\mu\nu}H_{\lambda\rho}$. Then if f is an embedding of  that sends the diagram's red part to a time t (and its green root to T), we have:

$\text{rvalue}_f($  $) = C_{\mu\nu}H_{\lambda\rho}\left(K^{T-t-1}\eta\right)^{\mu\lambda}\left(K^{T-t-1}\eta\right)^{\nu\rho}$

Here,  has two edges, which correspond in this example to the tensor contractions via $(K^{\cdots}\eta)^{\mu\lambda}$ and via $(K^{\cdots}\eta)^{\nu\rho}$, respectively. ◇

A.4.3. OVERALL

In sum, we evaluate an embedding of a diagram by using the **node**, **outline**, and **edge** rules to build an expression of $\nabla^d l_x$s, $\mathbb{E}_x$s and ηs. The difference between uvalues and rvalues lies only in their edge rule.

A.5. How to sum the embeddings' values

We give examples of automorphism groups and we illustrate the integration mentioned in this green summary box:

WE OBTAIN OVERALL final testing loss expressions by adding together the uvalues of all diagram embeddings, weighted by automorphism-group sizes as in 1.

If we are using rvalues instead of uvalues, we approximate sums over embeddings by integrals over t, we approximate $(I - \eta H)^t$ by $\exp(-\eta H t)$, and we apply:rule

$$\int_{0\leq u<T} du\ \exp(-uA) = (I - \exp(-TA))/A$$

When written in an eigenbasis of ηH, this A's coefficients are sums of one or more eigenvalues of ηH (one eigenvalue for each edge involved in the relevant degrees of freedom over which we integrate).

The Restated Key Lemma and Theorem 1 together say

Theorem *For any T: for η small enough, SGD has expected testing loss*

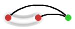
$$\sum_{\substack{D \text{ a linkless} \\ \text{diagram}}} \sum_{\substack{f \text{ an embed-} \\ \text{-ding of D}}} \frac{(-B)^{-|\text{edges}(D)|}}{\left|\text{Aut}_f(D)\right|} \text{rvalue}_f(D)$$

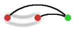
which is the same as

$$\sum_{\substack{D \text{ a} \\ \text{diagram}}} \sum_{\substack{f \text{ an embed-} \\ \text{-ding of D}}} \frac{(-B)^{-|\text{edges}(D)|}}{\left|\text{Aut}_f(D)\right|} \text{uvalue}(D)$$

Here, B is the batch size.

How do we evaluate the above sum? Summing uvalues reduces to counting embeddings, which in all the applications reported in this text is a routine combinatorial exercise. However, when summing rvalues, it is often convenient to replace a sum over embeddings by an integral over times, and the power $(I - \eta H)^{\Delta t - 1}$ by the exponential $\exp -\Delta t \eta H$. This incurs a term-by-term $1 + o(\eta)$ error factor, meaning that it preserves leading order results.

Example 4 Let us return to $D =$ , embedded, say, in the grid of one-epoch one-sample-per-batch SGD. From Example 3, we know that we want to sum the following value over all embeddings f, i.e. over all $0 \leq t < T$ to which the red part of the diagram's partition may be assigned:

rvalue$_f($  $) = C_{\mu\nu} \left(K^{T-t-1}\eta\right)^{\mu\lambda} \left(K^{T-t-1}\eta\right)^{\nu\rho} H_{\lambda\rho}$

Each embedding has a factor $(-B)^{-|\text{edges}(D)|} / \left|\text{Aut}_f(D)\right| = (-B)^{-2}/2$; we will multiply in this factor at the end so we now we focus on the $\sum_f$. So, using the aforementioned approximation, we seek to evaluate

$$\int_{0 \leq t < T} dt\, C_{\mu\nu} \left(\exp\left(-(T-t)\eta H\right)\eta\right)^{\mu\lambda} \left(\exp\left(-(T-t)\eta H\right)\eta\right)^{\nu\rho} H_{\lambda\rho} =$$
$$C_{\mu\nu} \left(\int_{0 \leq t < T} dt\ \exp\left(-(T-t)((\eta H) \otimes I + I \otimes (\eta H)\right)^{\mu\nu}_{\pi\sigma}\right) \eta^{\pi\lambda} \eta^{\sigma\rho} H_{\lambda\rho}$$

We know from linear algebra and calculus that $\int_{0 \leq u < T} du\ \exp(-uA) = (I - \exp(-TA))/A$ (when A is a non-singular linear endomorphism). Applying this rule for $u = T - t$ and $A = (\eta H) \otimes I + I \otimes (\eta H)$, we evaluate the integral as:

$$\cdots = C_{\mu\nu} \left(\frac{I - \exp\left(-T((\eta H) \otimes I + I \otimes (\eta H))\right)}{(\eta H) \otimes I + I \otimes (\eta H)}\right)^{\mu\nu}_{\pi\sigma} \eta^{\pi\lambda} \eta^{\sigma\rho} H_{\lambda\rho}$$

This is perhaps easier to write in an eigenbasis of ηH:

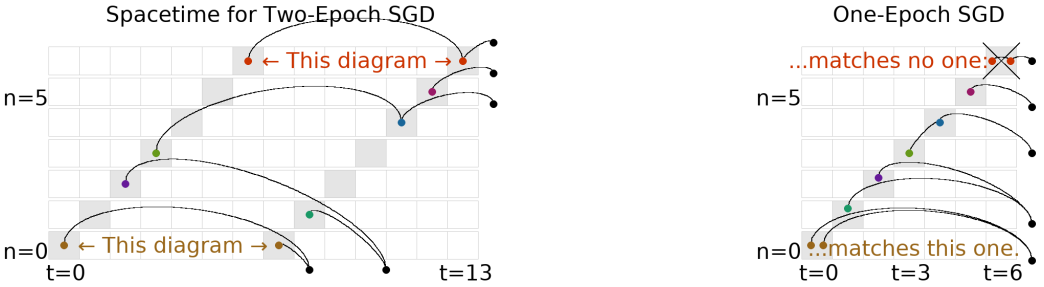
$$\cdots = \sum_{\mu\nu} C_{\mu\nu} \frac{1 - \exp\left(-T((\eta H)^{\mu}_{\mu} + (\eta H)^{\nu}_{\nu})\right)}{(\eta H)^{\mu}_{\mu} + (\eta H)^{\nu}_{\nu}} (\eta H \eta)^{\mu\nu}$$

Multiplying this expression by the aforementioned $(-B)^{-2}/2$ gives the contribution of to SGD's test loss. ◇

An **automorphism of D that preserves an embedding f of D** is a relabeling of D's nodes that preserves the root, the graph structure, the equivalence relation on non-root nodes, and that at the same time respects f: f must send a node to the same (n, t) pair to which the node's relabeling is sent. The sizes of automorphism groups appear in the denominators of our main results. They are usually of size one (that is, they usually contain only the the identity relabeling).

Example 5 (Automorphisms) We take as examples the diagram embeddings of §**??** (figure reproduced here). All the embeddings in the **left figure** have $|\mathrm{Aut}_f(D)| = 1$. For instance, the bottom-most embedding (of) has a non-trivial relabeling (namely, swap the two non-root nodes) that obeys all of the automorphism conditions *except* the "respects f" condition. Indeed, the two non-root nodes are assigned to different cells in the grid, so we may not swap them without violating the "respects f" condition.

By contrast, one of the embeddings (among the valid embeddings) shown in the **right figure** has a non-trivial automorphism group. This is the bottom-most embedding (of , again). Observe that swapping that diagram's two non-root nodes preserves the root, preserves the graph structure, and preserves the equivalence relation on non-root nodes. Moreover, such swapping respects f, since the two swapped nodes embed into the same cell. Thus, in this case, $|\mathrm{Aut}_f(D)| = 2$. ◇



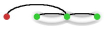
A.6. How to solve variant problems

In §B.7, we briefly discuss second-order methods and natural gradient descent. Here, we briefly discuss modifications. We omit proofs, which would closely follow §B's proof of the expectation-of-test-loss case.

VARIANCE (INSTEAD OF EXPECTATION)

To compute variances instead of expectations (with respect to the noise in the training set), one considers generalized diagrams that have "two roots" instead of one. More precisely, to compute, say, the un-centered second moment of testing loss, one uses diagrams whose edge structures are not rooted trees but instead forests consisting of two rooted trees. We require that the set of roots (now a set of size two instead of size one) is a part of the diagram's partition. We draw the two roots rightmost. For example, the generalized diagrams or may appear in this computation.

MEASURING ON THE TRAINING (INSTEAD OF TEST) SET

To compute the training loss, we compute with all the same diagrams as the testing loss, and we also allow all the additional generalized diagrams that violate the constraint that a diagram's root should be in a part of size one. Therefore, to compute the generalization gap (i.e. testing loss minus training loss), we sum over all the diagrams that expressly violate this constraint (and then, since gen. gp is test minus train instead of train minus test, we multiply the whole answer by -1). For example, the generalized diagrams or  may appear in this computation.

WEIGHT DISPLACEMENT (INSTEAD OF LOSS)

To compute displacements instead of losses, one considers generalized diagrams that have a "loose end" instead of a root. For example, the generalized diagrams or may appear in this computation.

A.7. Do diagrams streamline computation?

Diagram methods from Stueckelberg to Peierls have flourished in physics because they enable swift computations and offer immediate intuition that would otherwise require laborious algebraic manipulation. We demonstrate how our diagram formalism likewise streamlines analysis of descent by comparing direct perturbation[1] to the new formalism on two sample problems.

Aiming for a conservative comparison of derivation ergonomics, we lean toward explicit routine when using diagrams and allow ourselves to use clever and lucky simplifications when doing direct perturbation. For example, while solving the first sample problem by direct perturbation, we structure the SGD and GD computations so that the coefficients (that in both the SGD and GD cases are) called $a(T)$ manifestly agree in their first and second moments. This allows us to save some lines.

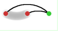
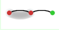
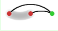
Despite these efforts, the diagram method yields arguments about *four times shorter* — and strikingly more conceptual — than direct perturbation yields. (We make no attempt to compare the re-summed version of our formalism to direct perturbation because the algebraic manipulations involved for the latter are too complicated to carry out.) These examples specifically suggest that: diagrams obviate the need for meticulous index-tracking, from the start focus one's attention on non-cancelling terms by making visually obvious which terms will eventually cancel, and allow immediate exploitation of a setting's special posited structure, for instance that we are initialized at a test minimum or that the batch size is 1. We regard these examples as evidence that diagrams offer a practical tool for the theorist.

We now compare Diagram Rules vs Direct Perturbation.

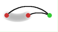
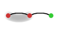
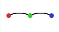
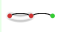
A.7.1. EFFECT OF BATCH SIZE

We compare the testing losses of pure SGD and pure GD. Because pure SGD and pure GD differ in how samples are correlated, their testing loss difference involves a covariance and hence occurs at order η^2.

DIAGRAM METHOD —

Since SGD and GD agree on noiseless landscapes, we consider only diagrams with fuzzy ties. Since we are working to second order, we consider only two-edged diagrams. There are only two such diagrams,  and . The first diagram, , embeds in GD's space time in N^2 as many

[1] By "direct perturbation", we mean direct application of our Key Lemma (§B.2).

ways as it embeds in SGD's spacetime, due to horizontal shifts. Likewise, there are N^2 times as many embeddings of  in distinct epochs of GD's spacetime as there are in distinct epochs of SGD's spacetime. However, each same-epoch embedding of  within any one epoch of GD's spacetime corresponds by vertical shifts to an embedding of  in SGD. There are $MN\binom{N}{2}$ many such embeddings in GD's spacetime, so GD's testing loss exceeds SGD's by $\frac{MN\binom{N}{2}}{N^2}$ . Reading the diagram's value from its graph structure, we unpack that expression as:

$$\eta^2 \frac{M(N-1)}{4} G\nabla C$$

DIRECT PERTURBATION —

We compute the displacement $\theta_T - \theta_0$ to order η^2 for pure SGD and separately for pure GD. Expanding $\theta_t \in \theta_0 + \eta a(t) + \eta^2 b(t) + o(\eta^2)$, we find:

$$\begin{aligned}
\theta_{t+1} &= \theta_t - \eta \nabla l_{n_t}(\theta_t) \\
&\in \theta_0 + \eta a(t) + \eta^2 b(t) - \eta(\nabla l_{n_t} + \eta \nabla^2 l_{n_t} a(t)) + o(\eta^2) \\
&= \theta_0 + \eta(a(t) - \nabla l_{n_t}) + \eta^2(b(t) - \nabla^2 l_{n_t} a(t)) + o(\eta^2)
\end{aligned}$$

To save space, we write l_{n_t} for $l_{n_t}(\theta_0)$. It's enough to solve the recurrence $a(t+1) = a(t) - \nabla l_{n_t}$ and $b(t+1) = b(t) - \nabla^2 l_{n_t} a(t)$. Since $a(0), b(0)$ vanish, we have $a(t) = -\sum_{0 \leq t < T} \nabla l_{n_t}$ and $b(t) = \sum_{0 \leq t_0 < t_1 < T} \nabla^2 l_{n_{t_1}} \nabla l_{n_{t_0}}$. We now expand l:

$$\begin{aligned}
l(\theta_T) &\in l + (\nabla l)(\eta a(T) + \eta^2 b(T)) \\
&\quad + \frac{1}{2}(\nabla^2 l)(\eta a(T) + \eta^2 b(T))^2 + o(\eta^2) \\
&= l + \eta((\nabla l) a(T)) + \eta^2((\nabla l) b(T) + \frac{1}{2}(\nabla^2 l) a(T)^2) + o(\eta^2)
\end{aligned}$$

Then $\mathbb{E}[a(T)] = -MN(\nabla l)$ and, since the N many singleton batches in each of M many epochs are pairwise independent,

$$\begin{aligned}
\mathbb{E}\left[(a(T))^2\right] &= \sum_{0 \leq t < T} \sum_{0 \leq s < T} \nabla l_{n_t} \nabla l_{n_s} \\
&= M^2 N(N-1) \mathbb{E}[\nabla l]^2 + M^2 N \mathbb{E}\left[(\nabla l)^2\right]
\end{aligned}$$

Likewise,

$$\begin{aligned}
\mathbb{E}[b(T)] &= \sum_{0 \leq t_0 < t_1 < T} \nabla^2 l_{n_{t_1}} \nabla l_{n_{t_0}} \\
&= \frac{M^2 N(N-1)}{2} \mathbb{E}\left[\nabla^2 l\right] \mathbb{E}[\nabla l] + \\
&\quad \frac{M(M-1)N}{2} \mathbb{E}\left[(\nabla^2 l)(\nabla l)\right]
\end{aligned}$$

Similarly, for pure GD, we may demand that a, b obey recurrence relations $a(t+1) = a(t) - \sum_n \nabla l_n/N$ and $b(t+1) = b(t) - \sum_n \nabla^2 l_n a(t)/N$, meaning that $a(t) = -t \sum_n \nabla l_n/N$ and $b(t) = \binom{t}{2} \sum_{n_0} \sum_{n_1} \nabla^2 l_{n_0} \nabla l_{n_1}/N^2$. So $\mathbb{E}\left[a(T)\right] = -MN(\nabla l)$ and

$$\begin{aligned}
\mathbb{E}\left[(a(T))^2\right] &= M^2 \sum_{n_0} \sum_{n_1} \nabla l_{n_0} \nabla l_{n_1} \\
&= M^2 N(N-1)\mathbb{E}\left[\nabla l\right]^2 + M^2 N \mathbb{E}\left[(\nabla l)^2\right]
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}\left[b(T)\right] = & \binom{MN}{2} \frac{1}{N^2} \sum_{n_0} \sum_{n_1} \nabla^2 l_{n_0} \nabla l_{n_1} \\
= & \frac{M(MN-1)(N-1)}{2} \mathbb{E}\left[\nabla^2 l\right] \mathbb{E}\left[\nabla l\right] + \\
& \frac{M(MN-1)}{2} \mathbb{E}\left[(\nabla^2 l)(\nabla l)\right]
\end{aligned}$$

We see that the expectations for a and a^2 agree between pure SGD and pure GD. So only b contributes. We conclude that pure GD's testing loss exceeds pure SGD's by

$$\begin{aligned}
& \eta^2 \left(\frac{M(MN-1)(N-1)}{2} - \frac{M^2 N(N-1)}{2} \right) \mathbb{E}\left[\nabla^2 l\right] \mathbb{E}\left[\nabla l\right]^2 \\
+ & \eta^2 \left(\frac{M(MN-1)N}{2} - \frac{M(M-1)N}{2} \right) \mathbb{E}\left[(\nabla^2 l)(\nabla l)\right] \mathbb{E}\left[\nabla l\right] \\
= & \eta^2 \frac{M(N-1)}{2} \mathbb{E}\left[\nabla l\right] \left(\mathbb{E}\left[(\nabla^2 l)(\nabla l)\right] - \mathbb{E}\left[\nabla^2 l\right] \mathbb{E}\left[\nabla l\right] \right)
\end{aligned}$$

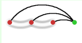
Since $(\nabla^2 l)(\nabla l) = \nabla((\nabla l)^2)/2$, we can summarize this difference as

$$\eta^2 \frac{M(N-1)}{4} G \nabla C$$

A.7.2. EFFECT OF NON-GAUSSIAN NOISE AT A MINIMUM.

We consider vanilla SGD initialized at a local minimum of the testing loss. One expects θ to diffuse around that minimum according to gradient noise. We compute the effect on testing loss of non-Gaussian diffusion. Specifically, we compare SGD testing loss on the loss landscape to SGD testing loss on a different loss landscape defined as a Gaussian process whose every covariance agrees with the original landscape's. We work to order η^3 because at lower orders, the Gaussian landscapes will by construction match their non-Gaussian counterparts.

DIAGRAM METHOD —

Because $\mathbb{E}[\nabla l]$ vanishes at initialization, all diagrams with a degree-one vertex that is a singleton vanish. Because we work at order η^3, we consider 3-edged diagrams. Finally, because all first and second moments match between the two landscapes, we consider only diagrams with at least one partition of size at least 3. The only such test diagram is . This embeds in T ways (one for each spacetime cell of vanilla SGD) and has symmetry factor $1/3!$ for a total of

$$\frac{T\eta^3}{6}\mathbb{E}\left[\nabla^3 l\right]\mathbb{E}\left[\nabla l_{n_{t_a}}\nabla l_{n_{t_b}}\nabla l_{n_{t_c}}\right]$$

DIRECT PERTURBATION —

We compute the displacement $\theta_T - \theta_0$ to order η^3 for vanilla SGD. Expanding $\theta_t \in \theta_0 + \eta a_t + \eta^2 b_t + \eta^3 c_t + o(\eta^3)$, we find:

$$\begin{aligned}
\theta_{t+1} &= \theta_t - \eta \nabla l_{n_t}(\theta_t) \\
&\in \theta_0 + \eta a_t + \eta^2 b_t + \eta^3 c_t \\
&\qquad - \eta\left(\nabla l_{n_t} + \nabla^2 l_{n_t}(\eta a_t + \eta^2 b_t) + \frac{1}{2}\nabla^3 l_{n_t}(\eta a_t)^2\right) + o(\eta^3) \\
&= \theta_0 + \eta\left(a_t - \nabla l_{n_t}\right) \\
&\qquad + \eta^2\left(b_t - \nabla^2 l_{n_t} a_t\right) \\
&\qquad + \eta^3\left(c_t - \nabla^2 l_{n_t} b_t - \frac{1}{2}\nabla^3 l_{n_t} a_t^2\right) + o(\eta^3)
\end{aligned}$$

We thus have the recurrences $a_{t+1} = a_t - \nabla l_{n_t}$, $b_{t+1} = b_t - \nabla^2 l_{n_t} a_t$, and $c_{t+1} = c_t - \nabla^2 l_{n_t} b_t - \frac{1}{2}\nabla^3 l_{n_t} a_t^2$ with solutions: $a_t = -\sum_t \nabla l_{n_t}$ and $\eta^2 b_t = +\eta^2 \sum_{t_0<t_1} \nabla^2 l_{n_{t_1}} \nabla l_{n_{t_0}}$. We do not compute c_t because we will soon see that it will be multiplied by 0. To third order, the testing loss of SGD is

$$\begin{aligned}
l(\theta_T) &\in l(\theta_0) + (\nabla l)(\eta a_T + \eta^2 b_T + \eta^3 c_T) \\
&\qquad + \frac{\nabla^2 l}{2}(\eta a_T + \eta^2 b_T)^2 \\
&\qquad + \frac{\nabla^3 l}{6}(\eta a_T)^3 + o(\eta)^3 \\
&= l(\theta_0) + \eta\left((\nabla l) a_T\right) \\
&\qquad + \eta^2\left((\nabla l) b_T + \frac{\nabla^2 l}{2} a_T^2\right) \\
&\qquad + \eta^3\left((\nabla l) c_T + (\nabla^2 l) a_T b_T + \frac{\nabla^3 l}{6} a_T^3\right) + o(\eta)^3
\end{aligned}$$

Because $\mathbb{E}[\nabla l]$ vanishes at initialization, we neglect the (∇l) terms. The remaining η^3 terms involve $a_T b_T$, and a_T^3. So let us compute their expectations:

$$\begin{aligned}
\mathbb{E}[a_T b_T] &= -\sum_{t}\sum_{t_0<t_1}\mathbb{E}\left[\nabla l_{n_t}\nabla^2 l_{n_{t_1}}\nabla l_{n_{t_0}}\right] \\
&= -\sum_{t_0<t_1}\sum_{t\notin\{t_0,t_1\}}\mathbb{E}[\nabla l_{n_t}]\,\mathbb{E}\left[\nabla^2 l_{n_{t_1}}\right]\mathbb{E}\left[\nabla l_{n_{t_0}}\right] \\
&\quad -\sum_{t_0<t_1}\sum_{t=t_0}\mathbb{E}\left[\nabla l_{n_t}\nabla l_{n_{t_0}}\right]\mathbb{E}\left[\nabla^2 l_{n_{t_1}}\right] \\
&\quad -\sum_{t_0<t_1}\sum_{t=t_1}\mathbb{E}\left[\nabla l_{n_t}\nabla^2 l_{n_{t_1}}\right]\mathbb{E}\left[\nabla l_{n_{t_0}}\right]
\end{aligned}$$

Since $\mathbb{E}[\nabla l]$ divides $\mathbb{E}[a_T b_T]$, the latter vanishes.

$$\begin{aligned}
\mathbb{E}\left[a_T^3\right] &= -\sum_{t_a,t_b,t_c}\mathbb{E}\left[\nabla l_{n_{t_a}}\nabla l_{n_{t_b}}\nabla l_{n_{t_c}}\right] \\
&= -\sum_{\substack{t_a,t_b,t_c\\ \text{disjoint}}}\mathbb{E}\left[\nabla l_{n_{t_a}}\right]\mathbb{E}\left[\nabla l_{n_{t_b}}\right]\mathbb{E}\left[\nabla l_{n_{t_c}}\right] \\
&\quad -3\sum_{t_a=t_b\neq t_c}\mathbb{E}\left[\nabla l_{n_{t_a}}\nabla l_{n_{t_b}}\right]\mathbb{E}\left[\nabla l_{n_{t_c}}\right] \\
&\quad -\sum_{t_a=t_b=t_c}\mathbb{E}\left[\nabla l_{n_{t_a}}\nabla l_{n_{t_b}}\nabla l_{n_{t_c}}\right]
\end{aligned}$$

As we initialize at a test minimum, only the last line remains, at it has T identical summands. When we plug into the expression for SGD testing loss, we get

$$\frac{T\eta^3}{6}\mathbb{E}\left[\nabla^3 l\right]\mathbb{E}\left[\nabla l_{n_{t_a}}\nabla l_{n_{t_b}}\nabla l_{n_{t_c}}\right]$$

Appendix B. Mathematics of the theory

B.1. Assumptions and Definitions

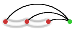
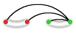
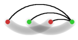
We assume throughout this work the following regularity properties of the loss landscape.

Existence of Taylor Moments — we assume that each finite collection of polynomials of the 0th and higher derivatives of the l_x, all evaluated at any point θ, may be considered together as a random variable insofar as they are equipped with a probability measure upon the standard Borel algebra.

Analyticity Uniform in Randomness — we assume that the functions $\theta \mapsto l_x(\theta)$ — and the expectations of polynomials of their 0th and higher derivatives — exist and are analytic with radii of convergence bounded from 0 (by a potentially θ-dependent function). So expectations and derivatives commute.

Boundedness of Gradients — we also assume that the gradients $\nabla l_x(\theta)$, considered as random covectors, are bounded by some continuous function of θ.[1] A metric-independent way of expressing this boundedness constraint is that the gradients all lie in some subset $\mathcal{S} \subseteq TM$ of the tangent bundle of weight space, where, for any compact $C \subseteq M$, we have that the topological pullback — of $\mathcal{S} \hookrightarrow TM \twoheadrightarrow M$ and $C \hookrightarrow M$ — is compact.

Now we turn to definitions.

Definition 4 (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 5 (Embedding a Diagram into a Grid) *An embedding of a diagram into a grid 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 they inhabit the same part of D's partition.*

We define $\text{uvalue}(D)$ and $\text{rvalue}_f(D)$ as in §A.4.

B.2. A key lemma à la Dyson

Suppose s is an analytic function defined on the space of weights. The following Lemma, reminiscent of Dyson (1949), helps us track $s(\theta)$ as SGD updates θ:

Key Lemma *For all T: for η sufficiently small, $s(\theta_T)$ is a sum over tuples of natural numbers:*

$$\sum_{(d_t:0\leq t<T)\in\mathbb{N}^T} (-\eta)^{\sum_t d_t} \left(\prod_{0\leq t<T} \left(\left.\frac{(g\nabla)^{d_t}}{d_t!}\right|_{g=\sum_{n\in\mathcal{B}_t} \nabla l_n(\theta)/B} \right)\right)(s)(\theta_0) \qquad (3)$$

Moreover, the expectation symbol (over training sets) commutes with the outer sum.

[1] Some of our experiments involve Gaussian noise, which is not bounded and so violates the hypothesis. In practice, Gaussians are effectively bounded in that our predictions vary smoothly with the first few moments of this distribution, so that a ±12-clipped Gaussian will yield almost the same predictions.

Here, we consider each $(g\nabla)^{d_t}$ as a higher order function that takes in a function f defined on weight space and outputs a function equal to the d_tth derivative of f, times g^{d_t}. The above product then indicates composition of $(g\nabla)^{d_t}$'s across the different t's. In total, that product takes the function s as input and outputs a function equal to some polynomial of s's derivatives.

Proof [Proof of the Key Lemma]We work in a neighborhood of the initialization so that the tangent space of weight space is a trivial bundle. For convenience, we fix a coordinate system, and with it the induced flat, non-degenerate inverse metric $\tilde{\eta}$; the benefit is that we may compare our varying η against one fixed $\tilde{\eta}$. Henceforth, a "ball" unless otherwise specified will mean a ball with respect to $\tilde{\eta}$ around the initialization θ_0. Since s is analytic, its Taylor series converges to s within some positive radius ρ ball. By assumption, every l_t is also analytic with radius of convergence around θ_0 at least some $\rho > 0$. Since gradients are x-uniformly bounded by a continuous function of θ, and since in finite dimensions the closed ρ-ball is compact, we have a strict gradient bound b uniform in both x and θ on gradient norms within that closed ball. When

$$2\eta T b < \rho\tilde{\eta} \tag{4}$$

as norms, SGD after T steps on any train set will necessarily stay within the ρ-ball.[1] We note that the above condition on η is weak enough to permit all η within some open neighborhood of $\eta = 0$.

Condition 4 together with analyticity of s then implies that $(\exp(-\eta g\nabla)s)\,(\theta) = s(\theta - \eta g)$ when θ lies in the $\tilde{\eta}$ ball (of radius ρ) and its η-distance from that $\tilde{\eta}$ ball's boundary exceeds b, and that both sides are analytic in η, θ on the same domain — and *a fortiori* when θ lies in the ball of radius $\rho(1 - 1/(2T))$. Likewise, a routine induction through T gives the value of s (after doing T gradient steps from an initialization θ) as

$$\left(\prod_{0\leq t<T} \exp(-\eta g\nabla)\Big|_{g=\nabla l_t(\theta)}\right)(s)(\theta)$$

for any θ in the $\rho(1 - T/(2T)$-ball (that is, the $\rho/2$-ball), and that both sides are analytic in η, θ on that same domain. Note that in each exponential, the ∇_ν does not act on the $\nabla_\mu l(\theta)$ with which it pairs.

Now we use the standard expansion of exp. Because (by analyticity) the order d coeffients of l_t, s are bounded by some exponential decay in d that has by assumption an x-uniform rate, we have absolute convergence and may rearrange sums. We choose to group by total degree:

$$\cdots = \sum_{0\leq d<\infty} (-\eta)^d \sum_{\substack{(d_t:0\leq t<T)\\ \sum_t d_t=d}} \left(\prod_{0\leq t<T} \frac{(g\nabla)^{d_t}}{d_t!}\Big|_{g=\nabla l_t(\theta)}\right) s(\theta) \tag{5}$$

The first part of the Key Lemma is proved. It remains to show that expectations over train sets commute with the above summation.

We will apply Fubini's Theorem. To do so, it suffices to show that

$$|c_d((l_t : 0\leq t<T))| \triangleq \left|\sum_{\substack{(d_t:0\leq t<T)\\ \sum_t d_t=d}} \left(\prod_{0\leq t<T} \frac{(g\nabla)^{d_t}}{d_t!}\Big|_{g=\nabla l_t(\theta)}\right) s(\theta)\right|$$

[1] The 2 ensures that SGD initialized at any point within a $\rho/2$ ball will necessarily stay within the ρ-ball.

has an expectation that decays exponentially with d. The symbol c_d we introduce purely for convenience; that its value depends on the train set we emphasize using function application notation. Crucially, no matter the train set, we have shown that the expansion 5 (that features c_d appear as coefficients) converges to an analytic function for all η bounded as in condition 4. The uniformity of this demanded bound on η implies by the standard relation between radii of convergence and decay of coefficients that $|c_d|$ decays exponentially in d at a rate uniform over train sets. If the expectation of $|c_d|$ exists at all, then, it will likewise decay at that same shared rate.

Finally, $|c_d|$ indeed has a well-defined expected value, for $|c_d|$ is a bounded continuous function of a (finite-dimensional) space of T-tuples (each of whose entries can specify the first d derivatives of an l_t) and because the latter space enjoys a joint distribution. So Fubini's Theorem applies. The Key Lemma follows. ■

B.3. From Dyson to diagrams

We now describe the terms that appear in the Key Lemma. The following result looks like Theorem 1, except it has $\mathrm{uvalue}(D)$ instead of $\mathrm{uvalue}_f(D)$, and the sum is over all diagrams, not just linkless ones. In fact, we will use Theorem 3 to prove Theorem 1.

Theorem 3 (Test Loss as a Path Integral) *For all T: for η sufficiently small, SGD's expected test loss is*

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

Here, D is a diagram whose root r does not participate in any fuzzy edge, f is an embedding of D into a grid, 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 (testing minus training loss).

Theorem 3 describe the terms that appear in the Key Lemma by matching each term to an embedding of a diagram in a grid, so that the infinite sum becomes a sum over all diagram grid configurations. The main idea is that the combinatorics of diagrams parallels the combinatorics of repeated applications of the product rule for derivatives applied to the expression in the Key Lemma. Balancing against this combinatorial explosion are factorial-style denominators, again from the Key Lemma, that we summarize in terms of the sizes of automorphism groups.

Proof [Proof of Theorem 3] We first prove the statement about testing losses. Due to the analyticity property established in our proof of the Key Lemma, it suffices to show agreement at each degree d and train set individually. That is, it suffices to show — for each train set $(l_n : 0 \leq n < N)$, grid S, function $\pi : S \to [N]$ that induces $\sim$, and natural d — that

$$(-\eta)^d \sum_{\substack{(d_t:0\leq t<T)\\ \sum_t d_t=d}} \left(\prod_{0\leq t<T} \left.\frac{(g\nabla)^{d_t}}{d_t!}\right|_{g=\nabla l_t(\theta)} \right) l(\theta) =$$

$$\sum_{\substack{D\in\mathrm{im}(\mathcal{F})\\ \text{with } d \text{ edges}}} \left(\sum_{f:D\to\mathcal{F}(S)} \frac{1}{\left|\mathrm{Aut}_f(D)\right|} \right) \frac{\mathrm{uvalue}_\pi(D,f)}{B^d} \tag{6}$$

Here, uvalue_π is the value of a diagram embedding before taking expectations over train sets. We have for all f that $\mathbb{E}\left[\mathrm{uvalue}_\pi(D, f)\right] = \mathrm{uvalue}(D)$. Observe that both sides of 6 are finitary sums.

Remark 2 (Differentiating Products) *The product rule of Leibniz easily generalizes to higher derivatives of finitary products:*

$$\nabla^{|M|}\prod_{k\in K} p_k = \sum_{\nu:M\to K}\prod_{k\in K}\left(\nabla^{\left|\nu^{-1}(k)\right|}p_k\right)$$

The above has $|K|^{|M|}$ many term indexed by functions to K from M.

We proceed by joint induction on d and S. The base cases wherein S is empty or $d = 0$ both follow immediately from the Key Lemma, for then the only embedding is the unique embedding of the one-node diagram •. For the induction step, suppose S is a sequence of $\mathcal{M} = \min S \subseteq S$ followed by a strictly smaller S and that the result is proven for $(\tilde d, \tilde S)$ for every $\tilde d \leq d$. Let us group by d_0 the terms on the left hand side of desideratum 6. Applying the induction hypothesis with $\tilde d = d - d_0$, we find that that left hand side is:

$$\sum_{0\leq d_0\leq d}\ \sum_{\substack{\tilde D\in\mathrm{im}(\mathcal{F})\\ \text{with } d-d_0 \text{ edges}}} \frac{1}{d_0!}\sum_{\tilde f:\tilde D\to\mathcal{F}(\tilde S)}\left(\frac{1}{\left|\mathrm{Aut}_{\tilde f}(\tilde D)\right|}\right)\cdot$$
$$(-\eta)^{d_0}\ (g\nabla)^{d_0}\Big|_{g=\nabla l_0(\theta)}\frac{\mathrm{uvalue}_\pi(\tilde D, \tilde f)}{B^{d-d_0}}$$

Since $\mathrm{uvalue}_\pi(\tilde D, \tilde f)$ is a multilinear product of $d-d_0+1$ many tensors, the product rule for derivatives tells us that $(g\nabla)^{d_0}$ acts on $\mathrm{uvalue}_\pi(\tilde D, \tilde f)$ to produce $(d-d_0+1)^{d_0}$ terms. In fact, $g = \sum_{m\in\mathcal{M}}\nabla l_m(\theta)/B$ expands to $B^{d_0}(d-d_0+1)^{d_0}$ terms, each conveniently indexed by a pair of functions $\beta : [d_0] \to \mathcal{M}$ and $\nu : [d_0] \to \tilde D$. The (β, ν)-term corresponds to an embedding f of a larger diagram D in the sense that it contributes $\mathrm{uvalue}_\pi(D, f)/B^{d_0}$ to the sum. Here, (f, D) is $(\tilde f, \tilde D)$ with $\left|(\beta\times\nu)^{-1}(n, \nu)\right|$ many additional edges from the cell of datapoint n at time 0 to the νth node of $\tilde D$ as embedded by $\tilde f$.

By the Leibniz rule of Remark , this (β, ν)-indexed sum by corresponds to a sum over embeddings f that restrict to $\tilde f$, whose terms are multiples of the value of the corresponding embedding of D. Together with the sum over $\tilde f$, this gives a sum over all embeddings f. So we now only need to check that the coefficients for each $f : D \to S$ are as claimed.

We note that the (β, ν) diagram (and its value) agrees with the $(\beta\circ\sigma, \nu\circ\sigma)$ diagram (and its value) for any permutation σ of $[d_0]$. The corresponding orbit has size

$$\frac{d_0!}{\prod_{(m,i)\in\mathcal{M}\times\tilde D}\left|(\beta\times\nu)^{-1}(m,i)\right|!}$$

by the Orbit Stabilizer Theorem of elementary group theory.

It is thus enough to show that

$$\left|\mathrm{Aut}_f(D)\right| = \left|\mathrm{Aut}_{\tilde f}(D)\right|\prod_{(m,i)\in\mathcal{M}\times\tilde D}\left|(\beta\times\nu)^{-1}(m,i)\right|!$$

We will show this by a direct bijection. First, observe that $f = \beta \sqcup \tilde f : [d_0]\sqcup\tilde D \to \mathcal{M}\sqcup\tilde S$. So each automorphism $\phi : D \to D$ that commutes with f induces both an automorphism $\mathcal{A} = \phi|_{\tilde D} : \tilde D \to \tilde D$

that commutes with $\tilde{f}$ together with the data of a map $\mathcal{B} = \phi_{[d_0]} : [d_0] \to [d_0]$ that both commutes with β. However, not every such pair of maps arises from a ϕ. For, in order for $\mathcal{A} \sqcup \mathcal{B} : D \to D$ to be an automorphism, it must respect the order structure of D. In particular, if $x \leq_D y$ with $x \in [d_0]$ and $y \in \tilde{D}$, then we need

$$\mathcal{B}(x) \leq_D \mathcal{A}(y)$$

as well. The pairs $(\mathcal{A}, \mathcal{B})$ that thusly preserve order are in bijection with the $\phi \in \mathrm{Aut}_f(D)$. There are $\left|\mathrm{Aut}_{\tilde{f}}(\tilde{D})\right|$ many $\mathcal{A}$. For each $\mathcal{A}$, there are as many $\mathcal{B}$ as there are sequences $(\sigma_i : i \in \tilde{D})$ of permutations on $\{j \in [d_0] : j \leq_D i\} \subseteq [d_0]$ that commute with $\mathcal{B}$. These permutations may be chosen independently; there are $\prod_{m \in \mathcal{M}} \left|(\beta \times \nu)^{-1}(m, i)\right|!$ many choices for σ_i. Claim **??** follows, and with it the correctness of coefficients.

The argument for generalization gaps parallels the above when we use $l - \sum_n l_n/N$ instead of l as the value for s. Theorem 3 is proved. ■

Remark 3 (The Case of $E = B = 1$ SGD) *The grid of $E = B = 1$ SGD permits all and only those embeddings that assign to each part of a diagram's partition a distinct cell. Such embeddings factor through a diagram ordering and are thus easily counted using factorials per Proposition 2. That proposition immediately follows from the now-proven Theorem 3.*

Proposition 2 *The order η^d contribution to the expected testing loss of one-epoch SGD with singleton batches is:*

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

where D ranges over d-edged diagrams. 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.

B.4. Interlude: a review of Möbius inversion

We say an embedding is **strict** if it assigns to each part a different datapoint n. Then, by Möbius inversion (Rota (1964)), a sum over strict embeddings of moment values (§A.4) matches a sum over all embeddings of uvalues.

B.5. Theorems 1 and 2

The diagrams summed in Theorem 1 and 2 may be grouped by their geometric realizations. Each nonempty class of diagrams with a given geometric realization has a unique element with minimally many edges, and in this way all and only linkless diagrams arise.

We encounter two complications: on one hand, that the sizes of automorphism groups might not be uniform among the class of diagrams with a given geometric realization. On the other hand, that the embeddings of a specific member of that class might be hard to count. The first we handle using Orbit-Stabilizer. The second we address as described by §B.4 via Möbius sums.

Proof [Proof of Theorem 1] We apply Möbius inversion (§B.4) to Theorem 3 (§B.3).

The difference in loss from the noiseless case is given by all the diagram embeddings with at least one fuzzy tie, where the fuzzy tie pattern is actually replaced by a difference between noisy and noiseless cases as prescribed by the preceding discussion on Möbius Sums. Beware that even relatively noiseless embeddings may have illegal collisions of non-fuzzily-tied nodes within a single grid (data) row. Throughout the rest of this proof, we permit such illegal embeddings of the fuzz-less diagrams that arise from the aforementioned decomposition.

Because the Taylor series for analytic functions converge absolutely in the interior of the disk of convergence, the rearrangement of terms corresponding to a grouping by geometric realizations preserves the convergence result of Theorem 3.

Let us then focus on those diagrams σ with a given geometric realization represented by an linkless diagram ρ. By Theorem 3, it suffices to show that

$$\sum_{f:\rho\to S}\ \sum_{\substack{\tilde{f}:\sigma\to S\\ \exists i_\star : f=\tilde{f}\circ i_\star}} \frac{1}{\left|\mathrm{Aut}_{\tilde{f}}(\sigma)\right|} = \sum_{f:\rho\to S}\ \sum_{\substack{\tilde{f}:\sigma\to S\\ \exists i_\star : f=\tilde{f}\circ i_\star}}\ \sum_{\substack{i:\rho\to\sigma\\ f=\tilde{f}\circ i}} \frac{1}{\left|\mathrm{Aut}_{f}(\rho)\right|} \tag{7}$$

Here, f is considered up to an equivalence defined by precomposition with an automorphism of ρ. We likewise consider $\tilde{f}$ up to automorphisms of σ. And above, i ranges through maps that induce isomorphisms of geometric realizations, where i is considered equivalent to $\hat{\imath}$ when for some automorphism $\phi \in \mathrm{Aut}_{\tilde{f}}(\sigma)$, we have $\hat{\imath} = i\circ\phi$. Name as X the set of all such is under this equivalence relation.

In equation 7, we have introduced redundant sums to structurally align the two expressions on the page; besides this rewriting, we see that equation 7's left hand side matches Theorem 3 resulting formula and tgat its right hand side is the desired formula of Theorem 1.

To prove equation 7, it suffices to show (for any $f, \tilde{f}, i$ as above) that

$$\left|\mathrm{Aut}_{f}(\rho)\right| = \left|\mathrm{Aut}_{\tilde{f}}(\sigma)\right| \cdot |X|$$

We will prove this using the Orbit Stabilizer Theorem by presenting an action of $\mathrm{Aut}_f(\rho)$ on X. We simply use precomposition so that $\psi \in \mathrm{Aut}_f(\rho)$ sends $i \in X$ to $i \circ \psi$. Since $f \circ \psi = f$, $i \circ \psi \in X$. Moreover, the action is well-defined, because if $i \sim \hat{\imath}$ by ϕ, then $i \circ \psi \sim \hat{\imath} \circ \psi$ also by ϕ.

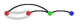
The stabilizer of i has size $\left|\mathrm{Aut}_{\tilde{f}}(\rho)\right|$. For, when $i \sim i \circ \psi$ via $\phi \in \mathrm{Aut}_{\tilde{f}}(\rho)$, we have $i \circ \psi = \phi \circ i$. This relation in fact induces a bijective correspondence: *every* ϕ induces a ψ via $\psi = i^{-1} \circ \phi \circ i$, so we have a map $\mathrm{stabilizer}(i) \hookleftarrow \mathrm{Aut}_{\tilde{f}}(\rho)$ seen to be well-defined and injective because structure set morphisms are by definition strictly increasing and because is must induce isomorphisms of geometric realizations. Conversely, every ψ that stabilizes enjoys *only* one ϕ via which $i \sim i \circ \phi$, again by the same (isomorphism and strict increase) properties. So the stabilizer has the claimed size.

Meanwhile, the orbit is all of $|X|$. Indeed, suppose $i_A, i_B \in X$. We will present $\psi \in \mathrm{Aut}_f(\rho)$ such that $i_B \sim i_A \circ \psi$ by $\phi = \text{identity}$. We simply define $\psi = i_A^{-1} \circ i_B$, well-defined by the aforementioned (isomorphisms and strict increase) properties. It is then routine to verify that $f \circ \psi = \tilde{f} \circ i_A \circ i_A^{-1} \circ i_B = \tilde{f} \circ i_B = f$. So the orbit has the claimed size, and by the Orbit Stabilizer Theorem, the coefficients in the expansions of Theorems 1 and 3 match. ■

Proof [Proof of Theorem 2] Since we assumed hessians are positive: for any m, the propagator $K^t = ((I - \eta H)^{\otimes m})^t$ exponentially decays to 0 (at a rate dependent on m). Since up to degree d only a finite number of diagrams exist and hence only a finite number of possible ms, the exponential rates are bounded away from 0. Moreover, for any fixed t_{big}, the number of diagrams — involving no exponent t exceeding t_{big} — is eventually constant as T grows. Meanwhile, the number involving at least one exponent t exceeding that threshold grows polynomially in T (with degree d). The exponential decay of each term overwhelms the polynomial growth in the number of terms, and the convergence statement follows. ■

B.6. Proofs of corollaries

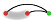
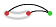
B.6.1. Corollary 4

Proof The relevant linkless diagram is  colorred (amputated as in the previous subsubsection). An embedding of this diagram into $E = B = 1$ SGD's grid is determined by two durations — t from red to green and $\tilde{t}$ from green to blue — obeying $t + \tilde{t} \leq T$. The automorphism group of each embedding has size 2: identity or switch the red nodes. So the answer is:

$$C_{\mu\nu} J^{\rho\lambda}_{\sigma} \left(\int_{t+\tilde{t}\leq T} (\exp(-t\eta H)\eta)^{\mu\rho} \, (\exp(-t\eta H)\eta)^{\nu\lambda} \, (\exp(-\tilde{t}\eta H)\eta)^{\sigma\pi} \right)$$

Standard calculus then gives the desired result. ■

B.6.2. Corollary 5's first part

Proof [Proof.] The relevant linkless diagram is  (which equals  because we are at a test minimum). This diagram has one embedding for each pair of same-row shaded cells, potentially identical, in a grid; for GD, the grid has every cell shaded, so each *non-decreasing* pair of durations in $[0, T]^2$ is represented; the symmetry factor for the case where the cells is identical is $1/2$, so we lose no precision by interpreting a automorphism-weighted sum over the *non-decreasing* pairs as half of a sum over all pairs. Each of these may embed into N many rows, hence the factor below of N. The two integration variables (say, $t, \tilde{t}$) separate, and we have:

$$\frac{N}{B^{\text{degree}}} \frac{C_{\mu\nu}}{2} \int_t (\exp(-t\eta H))^{\mu}_{\lambda} \int_{\tilde{t}} (\exp(-\tilde{t}\eta H))^{\nu}_{\rho} \, \eta^{\lambda\sigma} \eta^{\rho\pi} H_{\sigma\pi}$$

Since for GD we have $N = B$ and we are working to degree 2, the prefactor is $1/N$. Since $\int_t \exp(at) = (I - \exp(-aT))/a$, the desired result follows. ■

B.6.3. Corollary 5's second part

We apply the generalization gap modification (described in §A.6) to Theorem 1's result about testing losses.

Proof [Proof] The relevant linkless diagram is . This diagram has one embedding for each shaded cell of grid; for GD, the grid has every cell shaded, so each duration from 0 to T is represented. So the generalization gap is, to leading order,

$$+\frac{C_{\mu\nu}}{N} \int_t (\exp(-t\eta H))^{\mu}_{\lambda} \, \eta^{\lambda\nu}$$

Here, the minus sign from the gen-gap modification canceled with the minus sign from the odd power of $-\eta$. Integration finishes the proof. ■

B.6.4. COROLLARIES 2 AND 1

Corollary 2 and Corollary 1 follow from plugging appropriate values of M, N, B into the following proposition.

Proposition 3 *To order η^2, the testing loss of SGD — on N samples for $T = MN$ timesteps with batch size B dividing N and with any shuffling scheme — has expectation*

$$l - MNG_\mu G^\mu + MN\left(MN - \frac{1}{2}\right)G_\mu H^\mu_\nu G^\nu$$
$$+ MN\left(\frac{M}{2}\right)C_{\mu\nu}H^{\mu\nu} + MN\left(\frac{M - \frac{1}{B}}{2}\right)\left(\nabla_\mu C^\nu_\nu\right)G^\mu/2$$

Proof [of Proposition 3] To prove Proposition 3, we simply count the embeddings of the diagrams, noting that the automorphism groups are all of size 1 or 2. See Table 2. ■

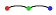
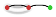
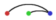
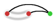
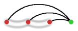
diagram	embed.s w/ $\lvert\mathrm{Aut}_f\rvert = 1$	embed.s w/ $\lvert\mathrm{Aut}_f\rvert = 2$
	1	0
	MNB	0
	$\binom{MN}{2}B^2$	0
	$N\binom{MB}{2}$	0
	$\binom{MNB}{2}$	MNB
	$N\binom{MB}{2}$	MNB

Table 2: Terms used in proof of Proposition 3

B.6.5. COROLLARY 3

The corollary's first part follows immediately from Proposition 3.

Proof [Proof of second part] Because $\mathbb{E}[\nabla l]$ vanishes at initialization, all diagrams with a degree-one vertex that is a singleton vanish. Because we work at order η^3, we consider 3-edged diagrams. Finally, because all first and second moments match between the two landscapes, we consider only diagrams with at least one part (in their partition) of size at least 3. The only such test diagram is . This embeds in T ways (one for each grid cell — recall that $E = B = 1$) and has symmetry factor $1/3!$ for a total of

$$\frac{T\eta^3}{6}\mathbb{E}\left[\nabla^3 l\right]\mathbb{E}\left[\nabla l_{n_{ta}}\nabla l_{n_{tb}}\nabla l_{n_{tc}}\right] = \frac{T\eta^3}{6}S_{\mu\nu\sigma}J^{\mu\nu\sigma}$$

This is the un-resummed expression. To obtain the re-summed expression, we replace $\eta^{\mu\nu}$ with $((I - \eta H)^{\Delta t - 1}\eta)^{\mu\nu}$. The embeddings range over T many times uniformly spaced in $[0, T]$. So we may integrate (let's name our variable of integration τ, and let's have it represent $\tau = \Delta t - 1 = T - t - 1$):

$$\int_{0 \leq \tau < T} (\exp(-\tau\eta H)\eta)^{\mu\nu}\,(\exp(-\tau\eta H)\eta)^{\pi\sigma}\,(\exp(-\tau\eta H)\eta)^{\lambda\rho}\,S_{\mu\pi\lambda}J_{\nu\sigma\rho}$$

Observe that

$$(\exp(-\tau\eta H))_{\mu}^{\nu}\,(\exp(-\tau\eta H))_{\pi}^{\sigma}\,(\exp(-\tau\eta H))_{\lambda}^{\rho} = (\exp(-\tau Y))_{\mu\nu\pi}^{\sigma\lambda\rho}$$

where $Y = \eta H \otimes I \otimes I + I \otimes \eta H \otimes I + I \otimes I \otimes \eta H$ is a six-index tensor. We finish by recalling that

$$\int_t \exp(At) = \frac{\exp(At)}{A}|_t$$

■

B.7. Future topics

Our diagrams invite exploration of Lagrangian formalisms and curved backgrounds:[1]

Question 3 *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.

Our work assumes a flat metric $\eta^{\mu\nu}$, but it might generalize to weight spaces curved in the sense of Riemann.[2] Such 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 and Kiela (2017)'s *hyperbolic embeddings*. While that work focuses on *optimization* on curved weight spaces, in machine learning we also wish to analyze *generalization*. Starting with the intuition that "smaller" hypothesis classes generalize better and that curvature controls the volume of small neighborhoods, we conjecture that sectional curvature regularizes learning:

Conjecture 1 (Sectional curvature regularizes) *If $\eta(\tau)$ is a Riemann metric on weight space, smoothly parameterized by τ, and if the sectional curvature through every 2-form at θ_0 increases as τ grows, then the gen. gap attained by fixed-T SGD with learning rate $c\eta(\tau)$ (when initialized from θ_0) decreases as τ grows, for all sufficiently small $c > 0$.*

We are optimistic our formalism may resolve conjectures such as above.

[1] Landau and Lifshitz (1960, 1951) review these concepts.

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

Appendix C. Experimental methods

C.1. What artificial landscapes did we use?

We define three artificial landscapes, called GAUSS, HELIX, and MEAN ESTIMATION.

C.1.1. GAUSS

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). More explicitly, the GAUSS landscape is a probability distribution $\mathcal{D}$ over functions $l_x : \mathbb{R}^1 \to \mathbb{R}$ on 1-dimensional weight space, indexed by standard-normally distributed 1-dimensional datapoints x and defined by the expression:

$$l_x(h) \triangleq \frac{1}{2}\left(h + x^2 \exp(-h)\right)$$

The gradient at sample x and weight σ is then $g_x(h) = (1 - x^2 \exp(-h))/2$. Since $x \sim \mathcal{N}(0, 1)$, the gradient $g_x(h)$ will be affinely related to a chi-squared, and in particular non-Gaussian.

To measure overfitting, we initialize at the true test minimum $h = 0$, then train and see how much the testing loss increases. At $h = 0$, the expected gradient vanishes, and the testing loss of SGD involves only diagrams that have no leaves of size one.

C.1.2. HELIX

The HELIX landscape has chirality, much like Archimedes' screw. Specifically, the HELIX landscape has weights $\theta = (u, v, z) \in \mathbb{R}^3$, data points $x \sim \mathcal{N}(0, 1)$, and loss:

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

Here,

$$H(\theta) = u^2 + v^2 + (\cos(z)u + \sin(z)v)^2$$

is quadratic in u, v, and

$$S(\theta) = \cos(z - \pi/4)u + \sin(z - \pi/4)v$$

is linear in u, v. Also, since $x \sim \mathcal{N}(0, 1)$, the $x \cdot S(\theta)$ term has expectation 0. In fact, the landscape has a three-dimensional continuous screw symmetry consisting of translation along z and simulateous rotation in the $u - v$ plane. Our experiments are initialized at $u = v = z = 0$, which lies within a valley of global minima defined by $u = v = 0$.

The paper body showed that SGD travels in HELIX' $+z$ direction. By topologically quotienting the weight space, say by identifying points related by a translation by $\Delta z = 200\pi$, we may turn the line-shaped valley into a circle-shaped valley. Then SGD eternally travels, say, counterclockwise. Alternatively, one may preserve the homotopy type of the underlying weight space by Nash-embedding a flat solid torus

$$[-10^1, +10^1] \times [-10^1, +10^1] \times [-10^3, +10^3]/((x, y, -10^3) \sim (x, y, +10^3))$$

in a higher-dimensional Euclidean space and extending HELIX from that torus to the ambient space.

Slightly modifying HELIX by adding a linear term $\alpha \cdot z$ to l for $\eta\alpha^2 \ll \eta^2/6$ leads SGD to perpetually ascend.

C.1.3. Mean Estimation

The Mean Estimation family of landscapes has 1 dimensional weights θ and 1-dimensional datapoints x. It is defined by the expression:

$$l_x(\theta) \triangleq \frac{1}{2} H\theta^2 + xS\theta$$

Here, H, S are positive reals parameterizing the family; they give the hessian and (square root of) gradient covariance, respectively.

For our hyperparameter-selection experiment (Figure 10 □□■) we introduce an l_2 regularization term as follows:

$$l_x(\theta, \lambda) \triangleq \frac{1}{2}(H + \lambda)\theta^2 + xS\theta$$

Here, we constrain $\lambda \geq 0$ during optimization using projections; we found similar results when parameterizing $\lambda = \exp(h)$, which obviates the need for projection but necessitates a non-canonical choice of initialization. We initialize $\lambda = 0$.

C.2. What image-classification landscapes did we use?

C.2.1. Architectures

In addition to the artificial loss landscapes Gauss, Helix, and Mean Estimation, we tested our predictions on logistic linear regression and simple convolutional networks (2 convolutional weight layers each with kernel 5, stride 2, and 10 channels, followed by two dense weight layers with hidden dimension 10) for the CIFAR-10 Krizhevsky (2009) and Fashion-MNIST datasets Xiao et al. (2017). The convolutional architectures used tanh activations and Gaussian Xavier initialization. To set a standard distance scale on weight space, we parameterized the model so that the Gaussian-Xavier initialization of the linear maps in each layer differentially pulls back to standard normal initializations of the parameters.

C.2.2. Datasets

For image classification landscapes, we regard the finite amount of available data as the true (sum of diracs) distribution $\mathcal{D}$ from which we sample testing and training sets in i.i.d. manner (and hence "with replacement"). We do this to gain practical access to a ground truth against which we may compare our predictions. One might object that this sampling procedure would cause testing and training sets to overlap, hence biasing testing loss measurements. In fact, testing and training sets overlap only in reference, not in sense: the situation is analogous to a text prediction task in which two training points culled from different corpora happen to record the same sequence of words, say, "Thank you!". In any case, all of our experiments focus on the limited-data regime, e.g. 10^1 datapoints out of $\sim 10^{4.5}$ dirac masses, so overlaps are rare.

C.3. Measurement process

C.3.1. Diagram evaluation on real landscapes

We implemented the formulae of §C.6 in order to estimate diagram values from real data measured at initialization from batch averages of products of derivatives.

C.3.2. Descent simulations

We recorded testing and training losses for each of the trials below. To improve our estimation of average differences, when we compared two optimizers, we gave them the same random seed (and hence the same training sets).

We ran $2 \cdot 10^5$ trials of Gauss with SDE and SGD, initialized at the test minimum with $T = 1$ and η ranging from $5 \cdot 10^{-2}$ to $2.5 \cdot 10^{-1}$. We ran $5 \cdot 10^1$ trials of Helixwith SGD with $T = 10^4$ and η ranging from 10^{-2} to 10^{-1}. We ran 10^3 trials of Mean Estimationwith GD and STIC with $T = 10^2$, H ranging from 10^{-4} to $4 \cdot 10^0$, a covariance of gradients of 10^2, and the true mean 0 or 10 units away from initialization.

We ran $5 \cdot 10^4$ trials of the CIFAR-10 convnet on each of 6 Glorot-Xavier initializations we fixed once and for all through these experiments for the optimizers SGD, GD, and GDC, with $T = 10$ and η between 10^{-3} and $2.5 \cdot 10^{-2}$. We did likewise for the linear logistic model on the one initialization of 0.

We ran $4 \cdot 10^4$ trials of the Fashion-MNIST convnet on each of 6 Glorot-Xavier initializations we fixed once and for all through these experiments for the optimizers SGD, GD, and GDC with $T = 10$ and η between 10^{-3} and $2.5 \cdot 10^{-2}$. We did likewise for the linear logistic model on the one initialization of 0.

C.4. Implementing optimizers

We approximated SDE by refining time discretization by a factor of 16, scaling learning rate down by a factor of 16, and introducing additional noise in the shape of the covariance in proportion as prescribed by the Wiener process scaling.

Our GDC regularizer was implemented using the unbiased estimator

$$\hat{C} \triangleq (l_x - l_y)_\mu l_{x\nu}/2$$

For our tests of regularization based on Corollary 5, we exploited the low-dimensional special structure of the artificial landscape in order to avoid diagonalizing to perform the matrix exponentiation: precisely, we used that, even on training landscapes, the covariance of gradients would be degenerate in all but one direction, and so we need only exponentiate a scalar.

C.5. Software frameworks and hardware

All code and data-wrangling scripts can be found on github.com/???????/perturb. This link will be made available after the period of double-blind review. Our code uses PyTorch 0.4.0 (Paszke et al., 2019) on Python 3.6.7; there are no other substantive dependencies. The code's randomness is parameterized by random seeds and hence reproducible. We ran experiments on a Lenovo laptop and on our institution's clusters; we consumed about 100 GPU-hours.

C.6. Unbiased estimators of landscape statistics

We use the following method — familiar to some of our colleagues but hard to find writings on — for obtaining unbiased estimates for various statistics of the loss landscape. The method is merely an elaboration of Bessel's factor (Gauss, 1823). For completeness, we explain it here.

Given samples from a joint probability space $\prod_{0 \leq d < D} X_d$, we seek unbiased estimates of *multipoint correlators* (i.e. products of expectations of products) such as $\langle x_0 x_1 x_2 \rangle \langle x_3 \rangle$. Here, angle brackets denote expectations over the population. For example, say $D = 2$ and from $2S$ samples we'd like to estimate $\langle x_0 x_1 \rangle$. Most simply, we could use $\mathsf{A}_{0\leq s<2S}\, x_0^{(s)} x_1^{(s)}$, where A denotes averaging over the sample. In fact, the following also works:

$$S \left(\underset{0\leq s<S}{\mathsf{A}}\, x_0^{(s)} \right) \left(\underset{0\leq s<S}{\mathsf{A}}\, x_1^{(s)} \right) + (1-S) \left(\underset{0\leq s<S}{\mathsf{A}}\, x_0^{(s)} \right) \left(\underset{S\leq s<2S}{\mathsf{A}}\, x_1^{(s)} \right) \tag{8}$$

When multiplication is expensive (e.g. when each $x_d^{(s)}$ is a tensor and multiplication is tensor contraction), we prefer the latter, since it uses $O(1)$ rather than $O(S)$ multiplications. This in turn allows more efficient use of batch computations on GPUs. We now generalize this estimator to higher-point correlators (and $D \cdot S$ samples).

For uniform notation, we assume without loss that each of the D factors appears exactly once in the multipoint expression of interest; such expressions then correspond to partitions on D elements, which we represent as maps $\mu : [D] \to [D]$ with $\mu(d) \leq d$ and $\mu \circ \mu = \mu$. Note that $|\mu| := |im(\mu)|$ counts μ's parts. We then define the statistic

$$\{x\}_\mu \triangleq \prod_{0\leq d<D} \underset{0\leq s<S}{\mathsf{A}}\, x_d^{(\mu(d)\cdot S+s)}$$

and the correlator $\langle x \rangle_\mu$ we define to be the expectation of $\{x\}_\mu$ when $S = 1$. In this notation, 8 says:

$$\langle x \rangle_{\boxed{0}\ \boxed{1}} = \mathbb{E}\left[S \cdot \{x\}_{\boxed{0\,1}} + (1-S) \cdot \{x\}_{\boxed{0}\ \boxed{1}} \right]$$

Here, the boxes indicate partitions of $[D] = [2] = \{0, 1\}$. Now, for general μ, we have:

$$\mathbb{E}\left[S^D \{x\}_\mu \right] = \sum_{\tau\leq\mu} \left(\prod_{0\leq d<D} \frac{S!}{\left(S - \left|\tau(\mu^{-1}(d))\right|\right)!} \right) \langle x \rangle_\tau \tag{9}$$

where '$\tau \leq \mu$' ranges through partitions *finer* than μ, i.e. maps τ through which μ factors. In smaller steps, 9 holds because

$$\begin{aligned}
\mathbb{E}\left[S^D \{x\}_\mu \right] &= \mathbb{E}\left[\sum_{(0\leq s_d<S)\in[S]^D} \prod_{0\leq d<D} x_d^{(\mu(d)\cdot S+s_d)} \right] \\
&= \sum_{\substack{(0\leq s_d<S) \\ \in[S]^D}} \mathbb{E}\left[\prod_{0\leq d<D} x_d^{\left(\min\left\{\tilde{d}\,:\,\mu(\tilde{d})\cdot S+s_{\tilde{d}}=\mu(d)\cdot S+s_d\right\}\right)} \right] \\
&= \sum_{\tau} \left| \left\{ \begin{matrix} (0\leq s_d<S) \in [S]^D : \\ \left(\begin{matrix} \mu(d)=\mu(\tilde{d}) \\ \wedge\ s_d=s_{\tilde{d}} \end{matrix} \right) \Leftrightarrow \tau(d)=\tau(\tilde{d}) \end{matrix} \right\} \right| \langle x \rangle_\tau \\
&= \sum_{\tau\leq\mu} \left(\prod_{0\leq d<D} \frac{S!}{\left(S - \left|\tau(\mu^{-1}(d))\right|\right)!} \right) \langle x \rangle_\tau
\end{aligned}$$

Solving 9 for $\langle x \rangle_\mu$, we find:

$$\boxed{\langle x \rangle_\mu = \frac{S^D}{S^{|\mu|}} \mathbb{E}\left[\{x\}_\mu \right] - \sum_{\tau<\mu} \left(\prod_{d\in im(\mu)} \frac{(S-1)!}{\left(S - \left|\tau(\mu^{-1}(d))\right|\right)!} \right) \langle x \rangle_\tau}$$

This expresses $\langle x\rangle_\mu$ in terms of the batch-friendly estimator $\{x\}_\mu$ as well as correlators $\langle x\rangle_\tau$ for τ *strictly* finer than μ. We may thus (use dynamic programming to) obtain unbiased estimators $\langle x\rangle_\mu$ for all partitions μ. Symmetries of the joint distribution and of the multilinear multiplication may further streamline estimation by turning a sum over τ into a multiplication by a combinatorial factor. For example, in the case of complete symmetry:

$$\langle x\rangle_{\boxed{012}} = S^2\{x\}_{\boxed{012}} - \frac{(S-1)!}{(S-3)!}\{x\}_{\boxed{0}\,\boxed{1}\,\boxed{2}} - 3\frac{(S-1)!}{(S-2)!}\{x\}_{\boxed{0}\,\boxed{12}}$$

C.7. Additional figures

In the rightmost figure, we add Corollary 5's generalization gap estimate to l. By descending on this regularized loss, we may tune smooth hyperparameters such as l_2 regularization coefficients for small datasets ($H \ll C/N$) (§C.7). Since matrix exponentiation takes time cubic in dimension, this regularizer is most useful for small models.

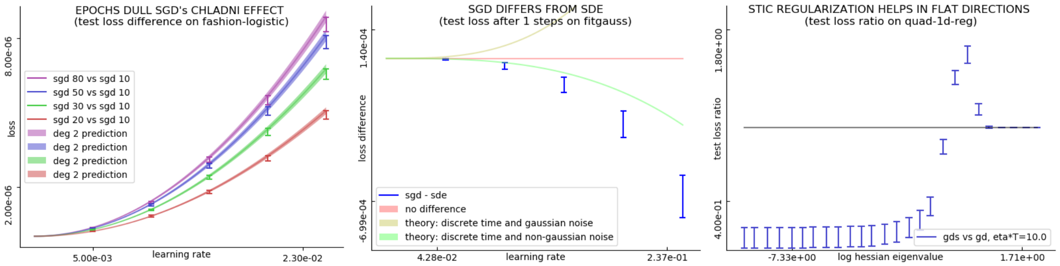


Figure 10: **Further experimental results**. **Left**: SGD with $2, 3, 5, 8$ epochs incurs greater test loss than one-epoch SGD (difference shown in I bars) by the predicted amounts (predictions shaded) for a range of learning rates. Here, all SGD runs have $N = 10$; we scale the learning rate for E-epoch SGD by $1/E$ to isolate the effect of inter-epoch correlations away from the effect of larger ηT. **Center**: SGD's difference from SDE after $\eta T \approx 10^{-1}$ with maximal coarseness on GAUSS. Two effects not modeled by SDE — time-discretization and non-Gaussian noise oppose on this landscape but do not completely cancel. Our theory approximates the above curve with a correct sign and order of magnitude; we expect that the fourth order corrections would improve it further. **Right**: Blue intervals show regularization using Corollary 5. When the blue intervals fall below the black bar, this proposed method outperforms plain GD. For MEAN ESTIMATION with fixed C and a range of Hs, initialized a fixed distance *away* from the true minimum, descent on an l_2 penalty coefficient λ improves on plain GD for most Hessians. The new method does not always outperform GD, because λ is not perfectly tuned according to STIC but instead descended on for finite ηT.