SGD at Small Learning Rates

Samuel C. Tenka   C O L I@M I T . E D U
MIT, CSAIL

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. Paired with results that relate overfitting to curvature, these repulsions suggest a mechanism for the unexpected generalization of overparameterized learners.

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

1. Introduction

1.1. Intuitions about SGD

One benefits from the intuition that stochastic gradient descent (SGD) approximates deterministic gradient descent (GD) (Bottou, 1991). This paper refines that intuition by showing how gradient noise biases learning toward certain areas of weight space.

Departing from prior work, we model discrete time and correlated gradient noise. Indeed, we derive corrections to continuous-time, independent noise approximations such as ordinary and stochastic differential equations (ODE, SDE). Our corrections lead to qualitative differences in dynamics: for example, we construct a non-pathological[1] loss landscape for which SGD's trajectory perpetually *circulates* counterclockwise; slightly modifying this landscape leads SGD to perpetually *ascend*. These examples, though artificial, demonstrate how drastically standard intuitions about SGD may fail. We argue that our theory's quantitative results enhance practical intuitions, and we verify our theory on convolutional CIFAR-10 and Fashion-MNIST loss landscapes.

At a high level, we Taylor-expand quantities of interest as functions of the learning rate η. We tame both the combinatorial explosion of terms and the large-time convergence difficulties by grouping terms according to the information-flow processes they represent. We introduce diagrams such as ⌢⌢, analogous to those of Feynman (1949) and Penrose (1971), to depict these processes.[2] We thus compute quantities such as expected testing losses by summing small sets of diagrams. Overall, our analysis offers and rigorously grounds a new physics-inspired perspective — of SGD as a superposition of many concurrent information-flow processes — that may inspire future work.

---

[1] All higher derivatives exist and are quadratically bounded; the gradient noise at each weight vector is 1-subgaussian.
[2] E.g. the diagram here shows a red update doubly affecting a green update that in turn affects the final testing loss. The diagrams with few edges suffice for leading order results. Colors help us refer to diagram parts; they lack meaning.

1.2. Background, notation, assumptions

We fix a *testing loss* function $l : \mathcal{M} \to \mathbb{R}$ on a space $\mathcal{M}$ of weights θ. We fix a distribution $\mathcal{D}$ from which unbiased estimates l_x of l are drawn. We write $(l_n : 0 \leq n < N)$ for a training sequence drawn i.i.d. from $\mathcal{D}$. 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. For instance, our theory accounts for tanh networks with cross entropy loss on bounded data — and with weight sharing, skip connections, soft attention, dropout, and weight decay. But it does not model ReLU networks.

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 $E = TB/N$ epochs or $M = T/N$ updates per point) of the form

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

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 shall especially study the *final testing loss* $\mathbb{E}[l(\theta_T)]$.

Our analysis makes heavy use of the tensors defined to the right: $G, H, J; C, S$ have $1, 2, 3; 2, 3$ indices, respectively. We shall 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 $C^\mu_\mu \triangleq \sum_{\mu\nu} \eta^{\mu\nu} \cdot C_{\nu\mu}$. A quantity q *vanishes to order* η^d when $\lim_{\eta \to 0} q/p(\eta) = 0$ for some homogeneous degree-d polynomial p; we then say $q \in o(\eta^d)$.

$$
\begin{aligned}
G &= \mathbb{E}_x[\nabla l_x(\theta)] &&= \\
H &= \mathbb{E}_x[\nabla\nabla l_x(\theta)] &&= \\
J &= \mathbb{E}_x[\nabla\nabla\nabla l_x(\theta)] &&= \\
C &= \mathbb{E}_x\left[(\nabla l_x(\theta) - G)^2\right] &&= \\
S &= \mathbb{E}_x\left[(\nabla l_x(\theta) - G)^3\right] &&=
\end{aligned}
$$

Above: Named tensors, typically evaluated at initialization ($\theta = \theta_0$). §2.3 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:* $\mathbb{E}[l(\theta_T) - l(\theta_0)] \in -TG_\mu G^\nu + o(\eta^1)$.

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. §2.3 details how evaluation of a *single diagram* gives the leading order result (Corollary 3), for concision quoted here assuming isotropic curvature ($\eta H \propto I$):

Proposition 1 *If we initialize near an 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 $J/\|\eta H\|$ 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 $18\|\eta H\|$.

1.3. Related work

It was Kiefer and Wolfowitz (1952) who, in uniting gradient descent (Cauchy, 1847) with stochastic approximation (Robbins and Monro, 1951), invented 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). Others approximate SGD by SDE to analyze implicit regularization (e.g. Chaudhari and Soatto (2018)), but, per Yaida (2019a), such continuous-time analyses cannot treat SGD noise correctly. 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 interpretation of the resulting terms offers a new qualitative picture of SGD as a superposition of several simpler information-flow processes.

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)), 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) are forced to assume uncorrelated Gaussian noise. By contrast, we use Penrose diagrams to compute testing losses to arbitrary order in η. We allow correlated, non-Gaussian noise and thus *any* smooth architecture. For instance, we do not assume information-geometric relationships between C and H,[1] so we may model VAEs.

2. Perturbative theory of SGD

2.1. An exegesis of a trivial example

We illustrate our approach by justifying Proposition 0. Intuitively, each step of SGD displaces θ by $-\eta G$ and 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 $\theta_T^\mu - \theta_0^\mu \in O(\eta^1)$ for each T. We claim that $\theta_T^\mu - \theta_0^\mu \in -T\eta^{\mu\nu}G_\nu + o(\eta^1)$. The claim holds when SGD is run for $T = 0$

---

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

timesteps. Moreover, if $T = T' + 1$ and the claim holds when SGD is run for T' timesteps, then:

$$\theta_T - \theta_{T'} = -\eta \nabla l_{T'}(\theta_{T'})$$
$$\in -\eta \nabla (l_{T'}(\theta_0) + \boxed{\nabla l_{T'}(\theta_0) \cdot (\theta_{T'} - \theta_0)} + o(\theta_{T'} - \theta_0))$$
$$\subseteq -\eta \nabla (l_{T'}(\theta_0) + \nabla l_{T'}(\theta_0) \cdot O(\eta^1) + o(O(\eta^1)))$$
$$\subseteq \boxed{-\eta \nabla l_{T'}(\theta_0)} + o(\eta^1)$$

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:

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

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 proof by keeping higher order terms. We emphasize three features that the above shares with its naïve generalizations: the "*transparency*" of future terms to old effects, the combinatorial *explosion* of intermediate terms, and the results' *polynomial* dependence on T.

By **transparency**, we mean that to first order the tth update contributes the same amount to θ_T no matter how much T exceeds t (transparency appears above in the constancy of $\boxed{-\eta \nabla l_{T'}(\theta_0)}$ with respect to θ_t for $t > 0$). Consequently, we may imagine each update as independently affecting the final test loss. This independence may be surprising, especially because it is false: SGD on a bounded landscape does not descend indefinitely at a constant rate. We thus wish to account for the landscape's curvature and hence to expand to higher order. Still, even for higher orders d of expansion, there is an analpgue of independence that finds precise expression in Theorem 1. The intuition that arises is that of many independent processes occuring, each blind to all but d updates.

When we expand to higher order, an **explosion** of terms rapidly obstructs progress. In a second order analysis, for instance, we may not discard the hessian term $\boxed{\eta \nabla \nabla l_{T'}(\theta_0) \cdot (\theta_{T'} - \theta_0)}$. The first-order 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, a second order analysis sums $O(T^2)$ terms, each (e.g. $\nabla \nabla l_{t=5} \nabla l_{t=2}$) involving a *pair* of times. In fact, the terms may vary in form: some (perhaps $\nabla \nabla l_{t=5} \nabla l_{t=2}$) of the quadratic terms that replace $\boxed{\mathbb{E}[\theta_T - \theta_0]}$ have statistically independent factors that permit expectations to factor, while others (perhaps $\nabla \nabla l_{t=5} \nabla l_{t=5}$) will 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, $\boxed{-T^1 G \eta G}$ 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 experience unbounded loss as T increases. By contrast, our main results (Theorems 1, 2) produce expressions that for each T have the correct Taylor data with respect to η and that for each

---

[1] There is no contradiction here: the domain and rate of convergence (of the Taylor series in η) typically depend on T, so we do not expect $\lim_{\eta \to 0}$ and $\lim_{T \to \infty}$ to commute.

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 of terms so that absolute convergence ensures consistency. The physical interpretation is to replace "*transparency*" by "*translucency*": we find that the strength of a term involving updates separated in time decays exponentially in the separation.

2.2. Perturbation as technique

We now abstractly describe the terms that appear when we keep higher order terms in our proof of Proposition 0. 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, 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:*

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

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

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.

2.3. Introducing diagrams

Though we here define diagrams, the following examples of valid and invalid diagrams by themselves provide a sufficient sense to read this paper's body.

Definition 1 (§B.3) *A **diagram** is a rooted tree equipped with a partition of its non-root nodes. We draw the tree structure using black edges, oriented left-to-right so that children precede parents. We depict the partition structure using minimally many gray outlined ties.*

valid diagrams	invalid diagrams
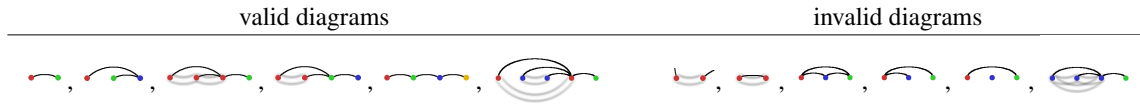
(One may draw dozens of small diagrams. In many analyses, all but a few diagrams will be irrelevant. For example, in the special case $E = B = 1$, each diagram with an ancestor-descendant pair in the same part evaluates to zero. When performing a fixed-T calculation to order η^d, we may neglect all diagrams with more than d edges. The $t^d T^{-p}$-th order correction[1] to the ordinary differential equation approximation of SGD is given by those diagrams with d edges and p many gray ties.)

---

[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 $p \neq 0$.

2.4. Insights from diagrams

2.5. Resummation

2.6. 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.

Theorem 1 (Special case of $E = B = 1$) *For any T: for η small enough, the final testing loss is*

$$\sum_{\substack{D \text{ an irreduc-} \\ \text{-ible diagram}}} \sum_{\substack{f \text{ an embed-} \\ \text{-ding of } D}} \frac{(-1)^{|\text{edges}(D)|}}{\left|\text{Aut}_f(D)\right|} \text{rvalue}_f(D)$$

Remark 1 *In practice, we approximate sums over embeddings by integrals over times and $(I-\eta H)^t$ by $\exp(-\eta Ht)$, 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 seen 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

3.1. Gradient noise repels SGD

Physical intuition suggests that noise repels SGD: if two neighboring regions of weight space have high and low levels of gradient noise, respectively, then we expect the rate at which θ jumps from the former to the latter to exceed the opposite rate. There is thus a net movement toward regions of small C.[1] Our theory makes this intuition precise; the drift is in the direction of $-\nabla C$, and the effect is strongest when gradient noise is not averaged out by large batch sizes:

Corollary 1 (⌣) *SGD with $E = B = 1$ avoids high-C regions more than GD: $l_C \triangleq \frac{N-1}{4N}\nabla^\mu C_\nu^\nu = \mathbb{E}[\theta_{GD} - \theta_{SGD}]^\mu - o(\eta^2)$. If $\hat{l}_c$ is a smooth unbiased estimator of l_c, 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.*

Roberts (2019) obtained a version of this with a nearly equal error of $O(\eta^2/N) \vee o(\eta^2)$

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 $M = 1$ and $\eta = \eta_0$ avoids high-C regions more than SGD with $M = M_0$ and $\eta = \eta_0/M_0$. Precisely: $\mathbb{E}[\theta_{M=M_0} - \theta_{M=1}]^\mu = \left(\frac{M_0-1}{4M_0}\right)N(\nabla^\mu C_\nu^\nu) + o(\eta^2)$.*

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.

---

[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 effect.

3.2. Non-Gaussian noise affects SGD but not SDE

Stochastic differential equations (SDE: see Liao et al. (2018)) are a popular theoretical approximation of SGD, but SDE and SGD differ 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, differences arise due to time discretization and non-Gaussian noise; the following corollary 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 unwieldy expression specializes to Proposition 1.

Intuitively, SGD and SDE respond differently to changes in curvature.

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*

$$-\frac{\eta^3}{2} \sum_{\mu\nu} C_{\mu\nu} \frac{1}{\eta(H_{\mu\mu} + H_{\nu\nu})} J_{\mu\nu\lambda} \frac{1}{H_{\lambda\lambda}} + o(\eta^2)$$
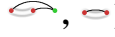
Intuitively, $D =$ ⌣ connects the subdiagram ⌣ $\propto CH$, via an extra edge on the green node (an extra ∇ on H), to D's degree-1 root, G. By l'Hôpital,[1] the displacement is $\propto -C\nabla H$. That is, SGD moves toward minima that are flat *with respect to C* (Figure 1 ■⊞). Taking limits to drop the non-degeneracy hypothesis, we expect *sustained* motion toward flat regions in a valley of minima. By avoiding Wei and Schwab (2019)'s assumptions of constant C, we find that SGD's velocity field is typically non-conservative, i.e. has curl (§4.2). Indeed, $\nabla(CH)$ is a total derivative but $C\nabla H$ is not. Since, by low-pass filter theory, $CH/2 + o(C)$ is the loss increase upon convolving l with a C-shaped Gaussian, 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 $\Theta(\eta^3)$. Indeed, our analysis integrates the noise over many updates, hence amplifying C's effect. Experiments verify our law.

So SGD descends on a C-smoothed landscape and prefers minima flat with respect to C.

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 1 □⊞). 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 (automatically) 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 $H \sim (\eta T)^{-1}$.

---

[1]Roughly: if a displacement $\Delta\theta$ grows loss by $GC\nabla H$ nats, and by G nats per foot, then $\Delta\theta$ is $C\nabla H$ feet.

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:*

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

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.
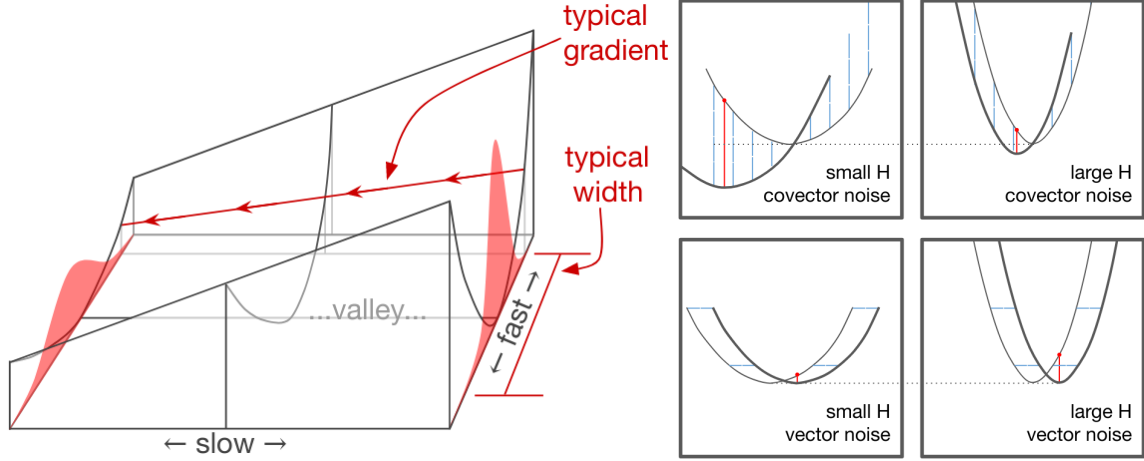
Figure 1: **Geometric intuition for curvature-noise interactions. Left**: Gradient noise pushes SGD toward flat minima (Corollary 4). Gray lines sketch a two-dimensional loss landscape near a valley of minima. The red densities show the typical θs, perturbed from the minimum due to noise C, in two cross sections of the loss valley. $J = \nabla H$ measures how curvature changes across the valley. Our theory does not assume separation between "fast" and "slow" modes, but we label them in the picture to ease comparison with Wei and Schwab (2019). Compare with Figure 3. **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).

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 2 ▦, Chaudhari and Soatto (2018) predict a velocity of 0 while we predict a velocity of $\eta^2/6$. Here, ⊥ bars, + signs, and shaded regions all mark 95% confidence intervals based on the standard error of the mean. §C describes neural architectures, of 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 order η^3 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 2 ⊞, ⊞). §C.7 supports Corollary 2's predictions about epoch number. Figure 2 ⊞ 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 2 ⊞).
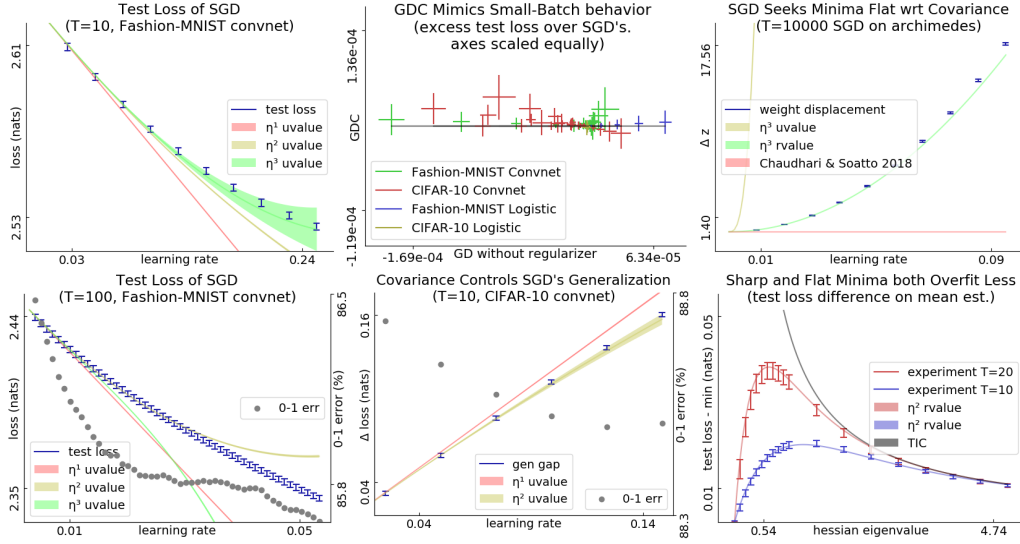
Figure 2: **Experiments on natural and artificial landscapes.** `rvalue` refers to Theorem 1's predictions, approximated as in Remark 1. `uvalue` indicates polynomial approximations to Theorem 1's result (see §A.5). `uvalues` are simpler to work with but (see ▦) may be less accurate.

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 and distinguishes GD from SGD. 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 travels HELIX' valley of global minima in the positive z direction. 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 TIC (Dixon and Ward, 2018) and thus on AIC.

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

To test the claimed dependence on C, §C.1 constructs a landscape, HELIX, with non-constant C throughout its valley of global minima. Figure 3 depicts HELIX' chiral shape.[1] As in Rock-Paper-Scissors, each point θ has a neighbor that, from $C(\theta)$'s perspective but not globally, is more attractive

---

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

(flatter). This permits eternal motion (into the page) despite the landscape's discrete translation symmetry in that direction.

More precisely, Corollary 4 predicts a *z*-velocity of $+\eta^2/6$ per timestep, while Chaudhari and Soatto (2018)'s SDE-based analysis predicts a constant velocity of 0.[1] Our prediction agrees with experiment (Figure 2 ▦). One may have sought an "effective loss" $\tilde{l}$ such that, up to $\sqrt{T}$ diffusion terms, SGD on l matches ODE on $\tilde{l}$. The non-conservativity of SGD's velocity shows that no such $\tilde{l}$ exists.
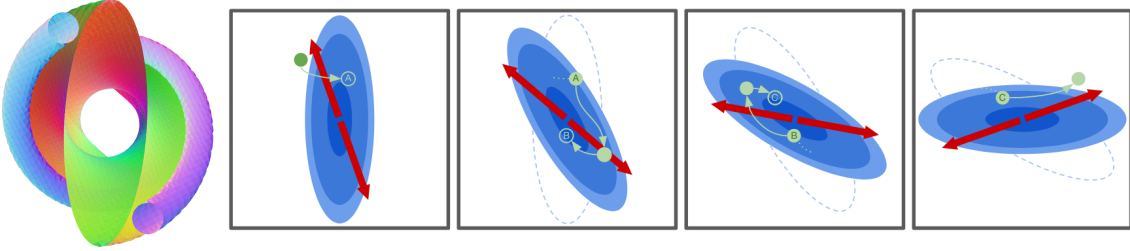
Figure 3: **Leftmost**: The HELIX landscape is defined on a three-dimensional weight space that extends indefinitely into and out of the page. A helical level surface (orange-green) of l winds around its axis, a one-dimensional valley of minima perpendicular to the page. l is large outside this surface. 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 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.

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

Prior work (§1.3) 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 2 ▦. To combat overfitting, we may 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.

---

[1]Indeed, HELIX' velocity is η-perpendicular to the image of $(\eta C)^\mu_\nu$ in tangent space.

5. Conclusion

We presented a diagram-based method for studying stochastic optimization on short timescales or near minima. Corollaries 4 and 5 together offer insight into SGD's success in training deep networks: SGD avoids curvature and noise, and curvature and noise 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, nonisotropic 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.

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.

Acknowledgments

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.

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.

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

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.

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.

M.P. Vitruvius. De architectura (book 10, chapter 6). *Self-published*, circa $10^{1/2}$ b.c.e.

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.

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.

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 space-time grid**, the batch size, training set size, and number of epochs.
- **Draw embeddings**, of diagrams into the space-time 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 an example calculation that follows 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. An example calculation: the effect of epochs

Question 1 *How does multi-epoch SGD differ from single-epoch SGD? Specifically, what is the difference between the expected 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 1's answer will be second-order (or higher-order) in η.

A.1.1. SPACE-TIME GRIDS

We take an $N \times T$ grid and shade its cells, shading the (n, t)th cell when the tth update involves the nth data point. Thus, each column contains B (batch size) many shaded cells and that each row contains E (epoch number) many shaded cells. The shaded grid is SGD's *space-time*. Two space-times are relevant to Question 1: one for multi-epoch SGD and another for single-epoch SGD — see Figure 4.
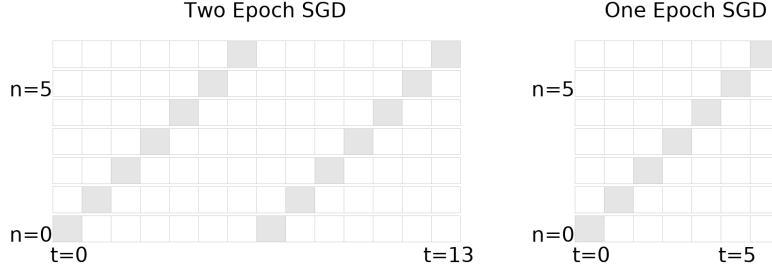
Figure 4: **The space-time grids of single-epoch and of multi-epoch SGD.** A cell at row n and column t is shaded provided that the nth training sample inhabits the tth batch. Both grids depict $N = 7$ training points and batch size $B = 1$; neither depicts training-set permutation between epochs. **Left**: 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. Embeddings of diagrams into space-time

There are four two-edged diagrams: ⌒, ⌒, ⌒, and ⌒. We permit the diagram ⌒, which violates the path condition mentioned in §**??**, because we are no longer restricting to the special case $E = B = 1$. An *embedding* of a diagram D into a space-time 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. Figure 5 shows some embeddings of order-1 and order-2 diagrams (i.e. one-edged and two-edged diagrams) into the space-times relevant to Question 1.
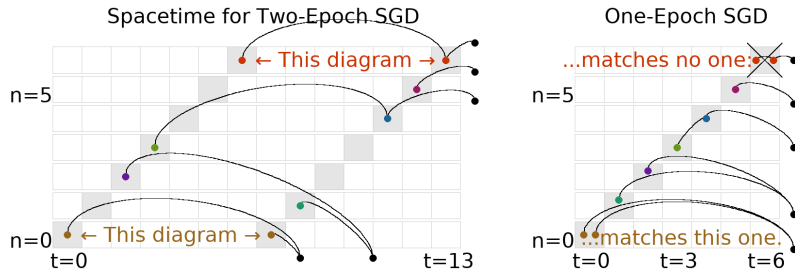

Figure 5: **The diagram ⌒ embeds into multi-epoch but not single-epoch space-time.** Drawn on each of the two grids are examples of embeddings. The black nodes external to the grids are positioned arbitrarily. From top to bottom in each grid, the five diagrams embedded are ⌒ (or ⌒), ⌒, ⌒, ⌒, and ⌒ (or ⌒). The diagram ⌒ may be embedded wherever the diagram ⌒ may be embedded, but not vice versa. Likewise for ⌒ and ⌒. **Left**: ⌒ embeds into multi-epoch space-time. **Right**: ⌒ cannot embed into single-epoch space-time. Indeed, the correlation condition forces both red nodes into the same row and thus the same cell, while the time-ordering condition forces the red nodes into distinct columns and thus distinct cells.

18

A.1.3. Values of the embeddings

We choose to compute uvalues instead of rvalues. The former are an approximation of the latter, appropriate when T is fixed instead of taken to infinity. uvalues have the same asymptotic error as rvalues with respect to η. Moreover, uvalues are simpler to calculate, since their numeric values depend only on diagrams, not on embeddings. So to compute a testing loss, we will we will multiply each diagram's uvalue by the number of ways that diagram embeds.

Figure 5 shows us that the diagram ⌢ embeds similarly into multi-epoch and single-epoch spacetimes. More precisely, 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 2-edged diagrams such as ⌢ scale as η^2, the total uvalue of the diagram's multi-epoch embeddings will match the total uvalue of the diagram's single-epoch embeddings. In fact, Figure 5 shows that this cancellation happens for all of the order-2 diagrams *except* for ⌒. Therefore, to second order, the answer to Question 1 will be (some multiple of) uvalue(⌒).

To compute ⌒'s value, we follow the rules in Section **??**; the edge rule for uvalues is that each edge becomes an η. So

$$\text{uvalue}(\text{⌒}) = \mathbb{E}\left[\nabla_\mu l_x \nabla_\nu \nabla_\lambda l_x\right] \mathbb{E}\left[\nabla_\rho l_x\right] \eta^{\mu\lambda} \eta^{\nu\rho} = (\nabla_\nu C_{\mu\lambda}/2) G^\rho \eta^{\mu\lambda} \eta^{\nu\rho}$$

A.1.4. Sum of the values

Referring again to Figure 5, we see that ⌒ has $\binom{M_0}{2} N$ many embeddings into the multi-epoch space-time (one embedding per pair of distinct epochs, per row) — and no embeddings into the single-epoch space-time. Moreover, each embedding of ⌒ has $\left|\text{Aut}_f(D)\right| = 1$. Now we plug into the overall formula for testing loss:

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

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 1 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.

Remark *An essentially similar argument proves Corollary 2.* ◇

Singleton Batches with Shuffling   Size-Two Batches without Shuffling
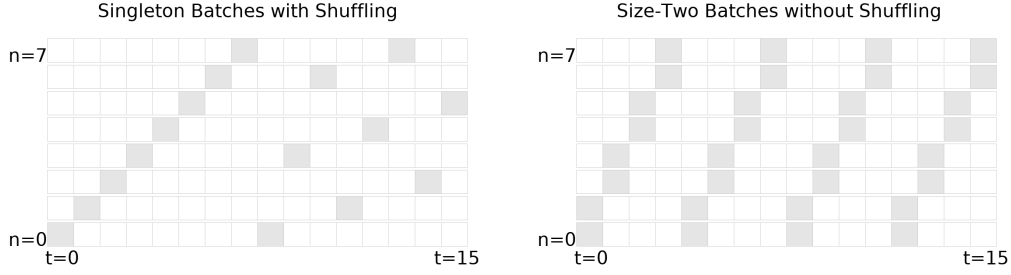

Figure 6: **The space-time grids of two SGD variants.** Shaded cells show (n, t) pairs (see text). **Left**: Two epoch SGD with batch size one. The training set is permuted between epochs. **Right**: Four epoch SGD with batch size two. The training set is not permuted between epochs.

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

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 *space-time grid* or *space-time*. See Figure 6.

In sum, 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.

Remark *A space-time grid also depicts the shuffling of training sets between epochs. Since each grid commits to a concrete sequence of training set permutations, we may analyze SGD with randomized permutations by taking expectations over multiple space-time grids. However, all of the corollaries in this text are invariant to inter-epoch training set permutations, so we will not focus on this point.[1]* ◇

---

[1]A routine check shows that for fixed T, inter-epoch shuffling yields only an $o(\eta^3)$ effect on testing losses.

A.3. How to identify the relevant diagram embeddings

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. Note that this definition generalizes the special case reported in the paper body; in particular, we no longer require the paper body's "path condition" to hold. 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 *irreducible* 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 irreducible. Only the first diagram in the list ⌢⌢, ⌢⌢⌢, ⋯ is irreducible.

Only the first diagram in the list ⌢⌢, ⌢⌢⌢, ⋯ is irreducible.

An *embedding* of a diagram D into a space-time 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 7.
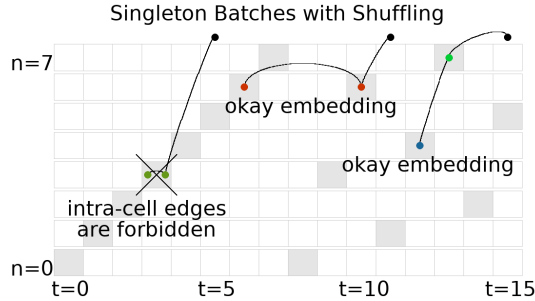
Figure 7: Embeddings, legal and illegal. **Left**: illegal embedding of ⌢⌢, since the time-ordering condition is not obeyed. For the same reason, not a legal embedding of ⌢⌢. **Middle**: an embedding of ⌢⌢. Also an embedding of ⌢⌢, since the correlation condition is obeyed. **Right**: a legal embedding 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 (Example ??) 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. Thus, the analysis in*

21

Example ?? was able to ignore diagrams such as ⌢. *Likewise, in §A.1 we argued by considering which embeddings that the only diagram relevant to Question 1 is* ⌢.   ◇

In sum, when using the diagram method to analyze how a quantity affects SGD to order $o(\eta^d)$, we must consider all diagrams with d or fewer edges that include that quantity as a component and that have a non-zero number of embeddings into the relevant space-time. If we are using rvalues (see next section for discussion of rvalues and uvalues), then we consider only the irreducible diagrams. For each diagram, we must enumerate the embeddings, i.e. the assignments of the diagram's nodes to space-time cells that obey both the time-ordering condition and correlation condition.

Here are some further examples. Table 1 shows the 6 diagrams that may embed into the space-time 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 space-time.

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

Table 1: **Multiple ways to draw the 6 distinct degree-3 diagrams for $B = E = 1$ SGD's testing loss.** Because the space-time 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 that obey §??'s path condition. 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 local minimum, rvalues do not diverge to $\pm\infty$ as $T \to \infty$.

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):

$$G \triangleq \mathbb{E}_x\left[\nabla l_x(\theta)\right] \triangleq \; \text{⌣}$$

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

$$H \triangleq \mathbb{E}_x\left[\nabla\nabla l_x(\theta)\right] \triangleq \; \text{⌄}$$

$$J \triangleq \mathbb{E}_x\left[\nabla\nabla\nabla l_x(\theta)\right] \triangleq \; \text{⌄}$$

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

$$\mathbb{E}_x\left[(\nabla l_x(\theta) - G)(\nabla\nabla l_x(\theta) - H)\right] \triangleq \; \text{⌣}$$

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

$$\mathbb{E}_x\left[(\nabla\nabla l_x(\theta) - H)(\nabla\nabla l_x(\theta) - H)\right] \triangleq \; \text{⌣}$$

$$\mathbb{E}_x\left[(\nabla l_x(\theta) - G)(\nabla\nabla\nabla l_x(\theta) - J)\right] \triangleq \; \text{⌣}$$

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

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 explained in §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. **Outline rule**: a partition on nodes evaluates to the difference $X - Y$, where X is the moment-value of the partition and Y is the sum of all strictly finer partitions.

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

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

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

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* $\text{⌣} = C_{\mu\nu}$ *and* $\text{⌄} = H_{\lambda\rho}$, *so that* $\text{⌣⌄} = C_{\mu\nu}H_{\lambda\rho}$. *Then*

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

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: 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 $\smile$ = $C_{\mu\nu}$ and $\curlyvee$ = $H_{\lambda\rho}$, so that $\smile\curlyvee = C_{\mu\nu}H_{\lambda\rho}$. Then if f is an embedding of $\frown$ that sends the diagram's red part to a time t (and its green root to T), we have:*

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

Here, $\frown$ 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

Theorem 1 in the paper body generalizes to

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

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

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 $D = \frown$, embedded, say, in the space-time of one-epoch one-sample-per-batch SGD. From Example 3, we know that we want to sum the following value over*

24

all embeddings f, i.e. over all $0 \leq t < T$ to which the red part of the diagram's partition may be assigned:

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

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

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

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

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:

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

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 $\overparen{\cdot\cdot\cdot\cdot}$ to SGD's test loss. ◇

In short, we sum embeddings of uvalues directly. We sum embeddings of rvalues using an integral-of-exponentials approximation along with the 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). As another example, see Example **??**.

A.6. Interpreting diagrams intuitively

We may intuitively interpret edges as carrying influence from the training set toward the test measurement. See Figure 9. From this perspective, we may intuitively interpret edges in an rvalue calculation as carrying influence from the training set toward the test measurement. See Figure 9.

A.7. 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.
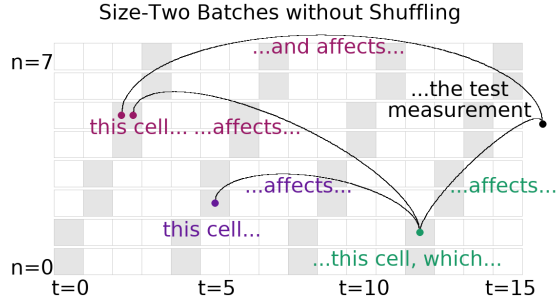
Figure 8: **Edges carry information**. Embedding of a 4-edged diagram.
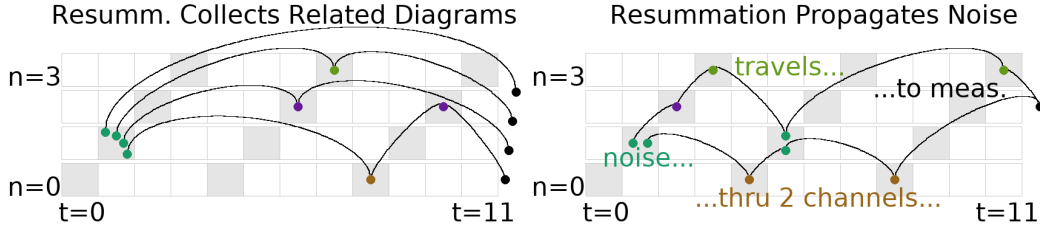
Figure 9: **Resummation propagates information, damped by curvature**. Each resummed value represents many un-resummed values, each modulated by the Hessian ($\smile$) in a different way. **Left**: Here is one of many un-resummed terms captured by a single resummed embedding for $\frown$. **Left**: each resummed value represents many un-resummed values. Here is one of many un-resummed terms captured by a single resummed embedding for $\frown$.

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. As in the case of test loss expectations, 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 $\frown$

or $\frown\frown$ 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 $\frown$ or $\frown\frown$ 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.8. 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 of argument.

    Despite these efforts, the diagram method yields arguments about *four times shorter* — and strikingly more conceptual — than direct perturbation yields. 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 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.

    We now compare Diagram Rules vs Direct Perturbation .

A.8.1. 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 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

---

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

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 $a(t) = -\sum_{0 \le t < T} \nabla l_{n_t}$ and $b(t) = \sum_{0 \le t_0 < t_1 < T} \nabla^2 l_{n_{t_1}} \nabla l_{n_{t_0}}$. 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,

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

Likewise,

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

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 $\mathbb{E}[a(T)] = -MN(\nabla l)$ and

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

and

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

. . .

$\cdots$ 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

$$\eta^2 \left(\frac{M(MN-1)(N-1)}{2} - \frac{M^2 N(N-1)}{2} \right) \mathbb{E}\left[\nabla^2 l\right] \mathbb{E}\left[\nabla l\right]^2$$

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

$$= \eta^2 \frac{M(N-1)}{2} \mathbb{E}\left[\nabla l\right] \left(\mathbb{E}\left[(\nabla^2 l)(\nabla l)\right] - \mathbb{E}\left[\nabla^2 l\right] \mathbb{E}\left[\nabla l\right] \right)$$

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.8.2. 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 $\mathbb{E}\left[\nabla l\right]$ 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:

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

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

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

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:

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

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

$$\mathbb{E}\left[a_T^3\right] = -\sum_{t_a, t_b, t_c} \mathbb{E}\left[\nabla l_{n_{t_a}} \nabla l_{n_{t_b}} \nabla l_{n_{t_c}}\right]$$

$$= -\sum_{\substack{t_a, t_b, t_c \\ \text{disjoint}}} \mathbb{E}\left[\nabla l_{n_{t_a}}\right] \mathbb{E}\left[\nabla l_{n_{t_b}}\right] \mathbb{E}\left[\nabla l_{n_{t_c}}\right]$$

$$- 3\sum_{t_a = t_b \neq t_c} \mathbb{E}\left[\nabla l_{n_{t_a}} \nabla l_{n_{t_b}}\right] \mathbb{E}\left[\nabla l_{n_{t_c}}\right]$$

$$- \sum_{t_a = t_b = t_c} \mathbb{E}\left[\nabla l_{n_{t_a}} \nabla l_{n_{t_b}} \nabla l_{n_{t_c}}\right]$$

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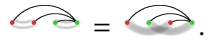
We assume throughout this work the following regularity properties of the loss landscape.

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 of 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 2 (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 3 (Embedding a Diagram into Spacetime) *An embedding of a diagram into a spacetime is an assignment of that diagram's non-root nodes to pairs (n, t) such that each node occurs at a time t' strictly after each of its children and such that two nodes occupy the same row n if they inhabit the same part of D's partition.*

We define $\mathrm{uvalue}(D)$ and $\mathrm{rvalue}_f(D)$ 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:*

$$\sum_{(d_t : 0 \leq t < T) \in \mathbb{N}^T} (-\eta)^{\sum_t d_t} \left(\prod_{0 \leq t < T} \left(\frac{(g\nabla)^{d_t}}{d_t!} \Bigg|_{g = \sum_{n \in \mathcal{B}_t} \nabla l_n(\theta)/B} \right) \right) (s)(\theta_0) \tag{2}$$

---

[1]Some of our experiments involve Gaussian noise, which is not bounded and so violates the hypothesis. In practice, Gaussians are effectively bounded, on the one hand in that with high probability no standard normal sample encountered on Gigahertz hardware within the age of the universe will much exceed $\sqrt{2 \log(10^{30})} \approx 12$, and on the other hand 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.

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

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

$$2\eta T b < \rho\tilde{\eta} \tag{3}$$

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 3 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

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

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:

$$\cdots = \sum_{0 \le d < \infty} (-\eta)^d \sum_{\substack{(d_t : 0 \le t < T) \\ \sum_t d_t = d}} \left(\prod_{0 \le t < T} \frac{(g\nabla)^{d_t}}{d_t!}\Big|_{g=\nabla l_t(\theta)}\right) s(\theta) \tag{4}$$

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

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

---

[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 4 (that features c_d appear as coefficients) converges to an analytic function for all η bounded as in condition 3. 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

TODO: define diagrams! FILL IN

We now describe the terms that appear in the Key Lemma. The following result looks like Theorem 1, except it has uvalue(D) instead of $\text{uvalue}_f(D)$, and the sum is over all diagrams, not just irreducible 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*

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

Here, D is a diagram whose root r does not participate in any fuzzy edge, f is an embedding of D into spacetime, and $\left|\text{Aut}_f(D)\right|$ counts the graph-automorphisms of D that preserve f's assignment of nodes to cells. If we replace D by $\left(-\sum_{p\in\text{parts}(D)}(D_{rp}-D)/N\right)$, 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 spacetime, so that the infinite sum becomes a sum over all diagram spacetime 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 $(l_n : 0 \le n < N)$, spacetime S, function $\pi : S \to [N]$ that induces $\sim$, and natural d — that

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

$$\sum_{\substack{D\in\text{im}(\mathcal{F}) \\ \text{with } d \text{ edges}}} \left(\sum_{f:D\to\mathcal{F}(S)} \frac{1}{\left|\text{Aut}_f(D)\right|}\right) \frac{\text{uvalue}_\pi(D,f)}{B^d} \tag{5}$$

Here, uvalue_π is the value of a diagram embedding before taking expectations over train sets. We have for all f that $\mathbb{E}[\text{uvalue}_\pi(D, f)] = \text{uvalue}(D)$. Observe that both sides of 5 are finitary sums.

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

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

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 $\cdot$. 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 5. Applying the induction hypothesis with $\tilde{d} = d - d_0$, we find that that left hand side is:

$$\sum_{\substack{0 \leq d_0 \leq d}} \sum_{\substack{\tilde{D} \in \text{im}(\mathcal{F}) \\ \text{with } d - d_0 \text{ edges}}} \frac{1}{d_0!} \sum_{\tilde{f}:\tilde{D} \to \mathcal{F}(\tilde{S})} \left(\frac{1}{\left|\text{Aut}_{\tilde{f}}(\tilde{D})\right|} \right) \cdot$$

$$(-\eta)^{d_0} \left. (g\nabla)^{d_0} \right|_{g = \nabla l_0(\theta)} \frac{\text{uvalue}_\pi(\tilde{D}, \tilde{f})}{B^{d - d_0}}$$

Since $\text{uvalue}_\pi(\tilde{D}, \tilde{f})$ 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 $\text{uvalue}_\pi(\tilde{D}, \tilde{f})$ 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 $v : [d_0] \to \tilde{D}$. The (β, v)-term corresponds to an embedding f of a larger diagram D in the sense that it contributes $\text{uvalue}_\pi(D, f)/B^{d_0}$ to the sum. Here, (f, D) is $(\tilde{f}, \tilde{D})$ with $\left|(\beta \times v)^{-1}(n, v)\right|$ many additional edges from the cell of datapoint n at time 0 to the vth node of $\tilde{D}$ as embedded by $\tilde{f}$.

By the Leibniz rule of Remark , this (β, v)-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, v) diagram (and its value) agrees with the $(\beta \circ \sigma, v \circ \sigma)$ diagram (and its value) for any permutation σ of $[d_0]$. The corresponding orbit has size

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

by the Orbit Stabilizer Theorem of elementary group theory.

It is thus enough to show that

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

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 $\phi \in \text{Aut}_f(D)$. There are $\left|\text{Aut}_{\tilde{f}}(\tilde{D})\right|$ 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 spacetime 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:*

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

where D ranges over d-edged diagrams. Here, D's parts have sizes $d_p : 0 \leq p \leq P$, and $|\text{ords}(D)|$ 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 irreducible 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 via Möbius sums.

Proof [Proof of Theorem 1] We apply Möbius inversion (§B.4) to Theorem 3 (§B.3). The result is that chains of embeddings FILL IN

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 spacetime (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 irreducible diagram ρ. By Theorem 3, it suffices to show that

$$\sum_{f:\rho\to S} \sum_{\substack{\tilde{f}:\sigma\to S \\ \exists i_\star : f = \tilde{f} \circ i_\star}} \frac{1}{\left|\text{Aut}_{\tilde{f}}(\sigma)\right|} = \sum_{f:\rho\to S} \sum_{\substack{\tilde{f}:\sigma\to S \\ \exists i_\star : f = \tilde{f} \circ i_\star}} \sum_{\substack{i:\rho\to\sigma \\ f=\tilde{f}\circ i}} \frac{1}{\left|\text{Aut}_f(\rho)\right|} \tag{6}$$

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 $\hat{i}$ when for some automorphism $\phi \in \text{Aut}_{\tilde{f}}(\sigma)$, we have $\hat{i} = i \circ \phi$. Name as X the set of all such is under this equivalence relation.

In equation 6, we have introduced redundant sums to structurally align the two expressions on the page; besides this rewriting, we see that equation 6'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 6, it suffices to show (for any $f, \tilde{f}, i$ as above) that

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

We will prove this using the Orbit Stabilizer Theorem by presenting an action of $\text{Aut}_f(\rho)$ on X. We simply use precomposition so that $\psi \in \text{Aut}_f(\rho)$ 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 $i \sim \hat{i}$ by ϕ, then $i \circ \psi \sim \hat{i} \circ \psi$ also by ϕ.

The stabilizer of i has size $\left|\text{Aut}_{\tilde{f}}(\rho)\right|$. For, when $i \sim i \circ \psi$ via $\phi \in \text{Aut}_{\tilde{f}}(\rho)$, 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 stabilizer$(i) \longleftrightarrow \text{Aut}_{\tilde{f}}(\rho)$ 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 $\psi \in \text{Aut}_f(\rho)$ such that $i_B \sim i_A \circ \psi$ by ϕ = 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

B.6. Proofs of corollaries

B.6.1. Corollary 4

Proof The relevant irreducible diagram is ⌢⌣ colorred (amputated as in the previous subsubsection). An embedding of this diagram into $E = B = 1$ SGD's spacetime 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 irreducible 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 spacetime; for GD, the spacetime 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.7) to Theorem 1's result about testing losses.

Proof [Proof] The relevant irreducible diagram is ⌣. This diagram has one embedding for each shaded cell of spacetime; for GD, the spacetime 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 M epochs with batch size B dividing N and with any shuffling scheme — has expectation*

$$l - MNG_\mu G^\mu + MN\left(MN - \frac{1}{2}\right)G_\mu H_\nu^\mu G^\nu$$

$$+ MN\left(\frac{M}{2}\right)C_{\mu\nu}H^{\mu\nu} + MN\left(\frac{M - \frac{1}{B}}{2}\right)\left(\nabla_\mu C_\gamma^\nu\right)G^\mu/2$$

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. Since we use fuzzy outlines instead of fuzzy ties, we allow untied nodes to occupy the same row, since the excess will be canceled out by the term subtract in the definition of fuzzy outlines. See Table **??**. ∎

diagram	embed.s w/ $\left\|\mathrm{Aut}_f\right\| = 1$	embed.s w/ $\left\|\mathrm{Aut}_f\right\| = 2$
.	1	0
⌢	MNB	0
⌢⌢	$\binom{MNB}{2}$	0
⌢	$N\binom{MB}{2}$	0
⌢	$\binom{MNB}{2}$	0
⌢	$N\binom{MB}{2}$	MNB

B.6.5. Corollary 3

The corollary's first part follows immediately from Proposition 3.
Proof [Proof of second part] Because $\mathbb{E}\left[\nabla l\right]$ 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) and has symmetry factor 1/3! for a total of

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

∎

B.7. Future topics

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

---

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

Question 2 *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.[1] 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] 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 its namesake's screw Vitruvius (circa $10^{1/2}$ b.c.e.). 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.

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 §**??** 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

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

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:

$$S \left(\underset{0 \leq s < S}{\mathsf{A}} x_0^{(s)} \right) \left(\underset{0 \leq s < S}{\mathsf{A}} x_1^{(s)} \right) + (1 - S) \left(\underset{0 \leq s < S}{\mathsf{A}} x_0^{(s)} \right) \left(\underset{S \leq s < 2S}{\mathsf{A}} x_1^{(s)} \right) \tag{7}$$

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

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

and the correlator $\langle x \rangle_\mu$ we define to be the expectation of $\{x\}_\mu$ when $S = 1$. In this notation, 7 says:

$$\langle x \rangle_{\boxed{0}\,\boxed{1}} = \mathbb{E}\left[S \cdot \{x\}_{\boxed{01}} + (1 - S) \cdot \{x\}_{\boxed{0}\,\boxed{1}} \right]$$

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

$$\mathbb{E}\left[S^D \{x\}_\mu \right] = \sum_{\tau \leq \mu} \left(\prod_{0 \leq d < D} \frac{S!}{\left(S - |\tau(\mu^{-1}(d))| \right)!} \right) \langle x \rangle_\tau \tag{8}$$

where '$\tau \leq \mu$' ranges through partitions *finer* than μ, i.e. maps τ through which μ factors. In smaller steps, 8 holds because

$$\mathbb{E}\left[S^D \{x\}_\mu \right] = \mathbb{E}\left[\sum_{(0 \leq s_d < S) \in [S]^D} \prod_{0 \leq d < D} x_d^{(\mu(d) \cdot S + s_d)} \right]$$

$$= \sum_{\substack{(0 \leq s_d < S) \\ \in [S]^D}} \mathbb{E}\left[\prod_{0 \leq d < D} x_d^{(\min\{\tilde{d} \,:\, \mu(\tilde{d}) \cdot S + s_{\tilde{d}} = \mu(d) \cdot S + s_d\})} \right]$$

$$= \sum_{\tau} \left| \left\{ \begin{array}{c} (0 \leq s_d < S) \in [S]^D \,: \\ \left(\begin{array}{c} \mu(d) = \mu(\tilde{d}) \\ \wedge \; s_d = s_{\tilde{d}} \end{array} \right) \Leftrightarrow \tau(d) = \tau(\tilde{d}) \end{array} \right\} \right| \langle x \rangle_\tau$$

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

Solving 8 for $\langle x \rangle_\mu$, we find:

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

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:

$$\langle x \rangle_{\boxed{012}} = S^2 \{x\}_{\boxed{012}} - \frac{(S-1)!}{(S-3)!} \{x\}_{\boxed{0}\,\boxed{1}\,\boxed{2}} - 3 \frac{(S-1)!}{(S-2)!} \{x\}_{\boxed{0}\,\boxed{12}}$$
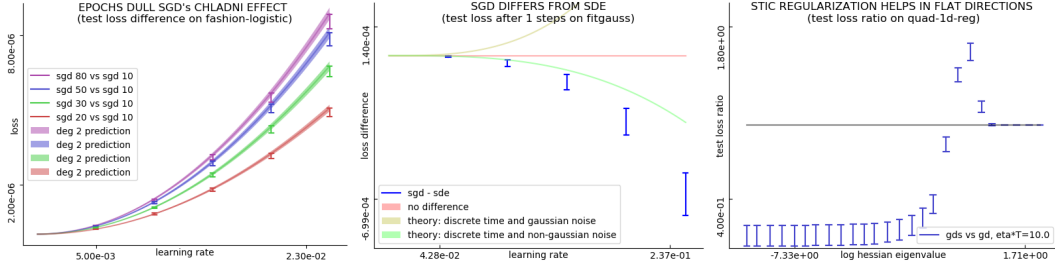
C.7. Additional figures

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 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.