A Perturbative Analysis of Stochastic Descent

Samuel C. Tenka

2020 July

Abstract. We analyze stochastic gradient descent (SGD) at small learning rates. Unlike prior analyses based on stochastic differential equations, our theory models discrete time and hence non-Gaussian noise. We illustrate our theory by discussing four of its corollaries: we (**A**) generalize the Akaike information criterion (AIC) to a smooth estimator of overfitting, hence enabling gradient-based model selection; (**B**) show how non-stochastic GD with a modified loss function may emulate SGD; (**C**) prove that gradient noise systematically pushes SGD toward flatter minima; and (**D**) characterize when and why flat minima overfit less than other minima.

Administrivia. I submit this thesis to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of **Master of Science** in Computer Science and Engineering at the Massachusetts Institute of Technology in the year 2020. I hereby grant to all organisms and institutions, including to MIT, permission to reproduce and to distribute publically paper and electronic copies of this document in whole or in part in any medium now known or hereafter created.

Accepted by: _____

1

Introduction

> This noise is extremely unpleasant. It sounds as if he was having an argument.
> I dislike arguments of any kind. They are always vulgar, and often convincing.
>
> Oscar Wilde

Users of deep learning benefit from the intuition that stochastic gradient descent (SGD) approximates noiseless gradient descent (GD). [†] This thesis refines that intuition by showing how gradient noise biases learning toward certain weights.

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

For example, we demonstrate that **noise** (the dependence of gradients on datapoints) interacts with **curvature** (the dependence of gradients on current weights) to push SGD toward flat minima, and we also explain when and why flatter minima overfit less. These results shed partial light on the surprising generalization properties observed in deep learning practice. Our thesis body highlights and intuitively motivates results such as these, leaving rigorous development to §B.

Beyond quantitative predictions, our analysis offers a novel interpretation of SGD as a sum of many concurrent interactions between weights and data. Diagrams such as ⌢, evocative of those of Feynman [1949] and Penrose [1971], depict these interactions. §4.2 discusses this bridge to physics — and its relation to Hessian methods and natural GD — as topics for future research.

1.1 Background on learning

Generalization, optimization, and approximation

This thesis studies generalization and optimization of neural networks. We first orient ourselves by reviewing in broad strokes a standard [‡] learning framework.

[‡] Y.S. Abu-Mostafa, M. Magdon-Ismail, and Hsuan-Tien Lin. Learning from data. *Caltech*, 2012

Learning means mapping a training sequence to a good hypothesis. Precisely, we posit a set $|\mathcal{D}|$ of **datapoints** equipped [§] with a probability distribution $\mathcal{D}$, a set C of **hypotheses**, and an **error function** $E : C \to |\mathcal{D}| \to \mathbb{R}$, which extends by averaging to $E : C \to |\mathcal{D}|^N \to \mathbb{R}$. For training data $\mathcal{S} \sim \mathcal{D}^N$, the training and test errors $\mathcal{E}_{\text{out}}, \mathcal{E}_{\text{in},\mathcal{S}} : C \to \mathbb{R}$ are $\mathcal{E}_{\text{out}} = \mathbb{E}_{\mathcal{D}} \circ E$ and $\mathcal{E}_{\text{in},\mathcal{S}} = \mathbb{E}_{\mathcal{S}} \circ E$. Observe that when $\theta \in C$ is independent from $\mathcal{S}$, $\mathcal{E}_{\text{in},\mathcal{S}}(\theta)$ is an unbiased estimator of $\mathcal{E}_{\text{out}}(\theta)$; the conclusion fails without independence, a phenomenon we know as **overfitting**.

[§] via a (partially defined) expectation

$$\mathbb{E} : (|\mathcal{D}| \to \mathbb{R}) \to \mathbb{R} \sqcup 1$$

We must later take care that expectations exist, but for now we work loosely.

A **learning rule** is then a map $\mathcal{L} : |\mathcal{D}|^N \to C$ from training sequences to hypotheses; [*] we seek low-error rules, that is, rules for which the test error

$$\mathbb{E}_{\mathcal{S}}(\mathcal{E}_{\text{out}} \circ \mathcal{L})$$

doesn't far exceed the best for which we dare to hope, namely the minimum training error $\mathcal{E}_{\text{in},\mathcal{S}}(\text{argmin}_C \mathcal{E}_{\text{in},\mathcal{S}})$. [†]

While C is typically uncountable, computers are finite; if for no other reason, [‡] we imagine $\mathcal{L}$'s range as some parametric "subset" $\mathcal{H}$ of practical hypotheses — more complicated to describe but easier to work with than C — with a canonical map $\mathcal{H} \to C$. [§] Moreover, in practice $\mathcal{L}$ takes the form of some (approximate, potentially regularized) optimization on the training error: $\mathcal{L}(\mathcal{S}) = \widehat{\text{argmin}}_{\mathcal{H}} (\mathcal{E}_{\text{in},\mathcal{S}})$. Therefore, we focus on $\widehat{\text{argmin}}_{\mathcal{H}} : (C \to \mathbb{R}) \to \mathcal{H}$ instead of on $\mathcal{L}$.

The problem of learning thus decomposes into the problems of **generalization**, **optimization**, and **approximation**:

$$\mathcal{E}_{\text{out}}(\widehat{\text{argmin}}_{\mathcal{H}} \mathcal{E}_{\text{in},\mathcal{S}}) - \mathcal{E}_{\text{in},\mathcal{S}}(\text{argmin}_C \mathcal{E}_{\text{in},\mathcal{S}}) \qquad \text{learning}$$
$$= \mathcal{E}_{\text{out}}(\widehat{\text{argmin}}_{\mathcal{H}} \mathcal{E}_{\text{in},\mathcal{S}}) - \mathcal{E}_{\text{in},\mathcal{S}}(\widehat{\text{argmin}}_{\mathcal{H}} \mathcal{E}_{\text{in},\mathcal{S}}) \qquad \text{generalization}$$
$$+ \mathcal{E}_{\text{in},\mathcal{S}}(\widehat{\text{argmin}}_{\mathcal{H}} \mathcal{E}_{\text{in},\mathcal{S}}) - \mathcal{E}_{\text{in},\mathcal{S}}(\text{argmin}_{\mathcal{H}} \mathcal{E}_{\text{in},\mathcal{S}}) \qquad \text{optimization}$$
$$+ \mathcal{E}_{\text{in},\mathcal{S}}(\text{argmin}_{\mathcal{H}} \mathcal{E}_{\text{in},\mathcal{S}}) - \mathcal{E}_{\text{in},\mathcal{S}}(\text{argmin}_C \mathcal{E}_{\text{in},\mathcal{S}}) \qquad \text{approximation}$$

Intuitively, $\mathcal{D}$ controls generalization, $\mathcal{L}$ controls optimization, and $\mathcal{H}$ controls approximation. [¶] To the extent this intuition holds, it permits disentangled analysis of the three separate concerns. In reality, the three concerns interact, for instance because the approximation term encourages large hypothesis classes $\mathcal{H}$, stymieing generalization and complicating optimization.

Neural networks

Deep learning partially solves the problem of learning by offering a class of $\mathcal{H}$'s with small approximation error and a heuristically motivated $\mathcal{L}$ that in practice optimizes well. In particular, **neural networks** are smoothly parameterized hypothesis classes for the supervised setting where $|\mathcal{D}| = \mathcal{I} \times \mathcal{Y}$ and $C = \mathcal{Y}^{\mathcal{I}}$. In practice, neural networks arise as compositions of simple functions that, via dynamic programming, afford efficient computation of parameter-output Jacobians. More precisely, we posit that $\mathcal{H}, \mathcal{Y}$ are smooth manifolds, and that the canonical map $\mathcal{H} \to C$ is given by an **architecture** written in curried forms as $A : \mathcal{I} \to \mathcal{H} \to \mathcal{Y}$ or $\tilde{A} : \mathcal{H} \to \mathcal{I} \to \mathcal{Y}$, where each $A(z)$ is smooth. [‖] We also posit an objective funcion written in curried form as $O : |\mathcal{D}| \to \mathcal{Y} \to \mathbb{R}$, with each $O(x)$ smooth. Then the error function (restricted to $\mathcal{H}$) is $E(\theta)(x) = O(x) \circ \tilde{A}(\theta)$ for $\theta \in \mathcal{H}$ and $x \sim \mathcal{D}$. The error function $E : \mathcal{H} \to |\mathcal{D}| \to \mathbb{R}$ alternatively curries to a **loss landscape** $l : |\mathcal{D}| \to \mathcal{H} \to \mathbb{R}$, which we regard as parameterizing a random function $l_x : \mathcal{H} \to \mathbb{R}$. [**] In short:

Symbol	$\mathcal{D}$	$\mathcal{H}$	$l_x : \mathcal{H} \to \mathbb{R}$
Concept	data distribution	hypotheses	loss at $x \in |\mathcal{D}|$

[*] Real learning rules are often randomized algorithms, not pure functions.

[†] The careful will use infima, not minima. But this section is a poetic sketch.

[‡] Other reasons include control over generalization and optimization.

[§] We permit the map to be non-injective. This is often the case in deep learning, wherein C is a space of measurable functions, $\mathcal{H}$ is a finite-dimensional space of weights, and certain permutations of a weight's components yields the same function.

[¶] This sentence pains the author's Bayesian sensibilities, but the fact is that this thesis is quite frequentist.

[‖] A typical $\tilde{A}(\theta)$ looks like

$$a_{100} \circ f_{99}(\theta) \circ a_{98} \circ \cdots \circ f_1(\theta) \circ a_0$$

with a, f smooth and f bilinear.

[**] Caution: we will overload notation by using l to mean l_x's expectation.

Gradient descent

By construction, each l_x is smooth, so we may implement $\widehat{\mathrm{argmin}}_{\mathcal{H}}$ by gradient descent. Explicitly, we choose an initial point $\theta_0 \in \mathcal{H}$ and then repeatedly **update** using learning rate η:

$$\theta_{t+1} = \theta_t - \eta(\nabla \mathcal{E}_{\mathrm{in},\mathcal{S}})(\theta_t)$$

Observe that θ takes values in $\mathcal{H}$ while $(\nabla \mathcal{E}_{\mathrm{in},\mathcal{S}})(\theta)$ takes values in the cotangent bundle $T^{\star}\mathcal{H}$. The above update is thus not yet well-defined. To bridge the two types, we interpret η as a map $\eta : T^{\star}\mathcal{H} \to T\mathcal{H}$ from covectors to vectors. That is, η has the type signature of an (inverse) metric. [*] Then $-\eta(\nabla \mathcal{E}_{\mathrm{in},\mathcal{S}})(\theta)$ is a vector, and we may flow along this vector geodesically from θ_t to θ_{t+1}.

[*] We assume a flat metric until §4.2, where we briefly discuss curvature.

The special case of linear image classification illustrates the metric's role in learning. Here, $\mathcal{I}$ is a vector space [†] of images and $\mathcal{H} = \mathcal{I}^{\star}$ is the space of linear features. Since $\mathcal{H}$ and $\mathcal{I}$ are dual, we may regard whatever inverse metric on $\mathcal{H}$ we use for descent as a metric on $\mathcal{I}$. If, intuitively, each feature is a question that can be asked of an image, then the metric $\eta : \mathcal{I} \to \mathcal{I}^{\star}$ turns an image z itself into a question, namely the question of "is this image like z?" Crucially, different choices of η will yield different questions for the same z — ranging from

[†] All our spaces are finite in dimension unless obviously otherwise.

is this image like z in that both are blue near the center?

(for a center-masked pixel-wise metric)

to

is this image like z in that their edges are aligned?

(for a Sobelev-inspired grayscale metric)

— and will thus lead to different generalization behaviors.

In sum, complete Riemannian manifolds are the natural setting for gradient descent. [‡] Many subfields of computing science have recognized the importance of the choice of metric, so the same circle of ideas takes the varied names of *learning on manifolds* [Bonnabel, 2013], *Legendre duality*, *the kernel trick*, *matrix pre-conditioning*, and *Galois connections*. To understand and exploit this choice of metric is a theme of this thesis.

[‡] much as symplectic spaces are the natural setting for classical mechanics

Stochastic gradient descent

Ideally, we would descend on $\mathcal{E}_{\mathrm{out}}$ rather than the estimator $\mathcal{E}_{\mathrm{in},\mathcal{S}} = \sum_{x \in \mathcal{S}} l_x/|\mathcal{S}|$ or more generally $l_{\mathcal{B}} = \sum_{x \in \mathcal{B}} l_x/|\mathcal{B}|$ for non-empty **batches** $\mathcal{B} \subseteq \mathcal{S}$. But we have access only to estimators, and we are confronted with a choice of which to use. This problem of batching represents a trade-off in optimization: while each large-batch update more precisely decreases the objective, we may perform more small-batch updates per unit time. In practice, one often samples batches of small, fixed size $|\mathcal{B}| = B \ll N$ from $\mathcal{S}$ for each update in **stochastic gradient descent** (SGD).

Though optimization efficiency motivates SGD, we will show that stochasticity biases learning away from certain regions of weight space, and we discuss when and why this helps generalization. Overall, we find that SGD differs from descent on $\mathcal{E}_{\mathrm{in},\mathcal{S}}$, not only through diffusion terms sublinear $(\sqrt{T})$ in the number T of updates but also through a linear drift that scales with T.

1.2 Example of diagram-based computation of SGD's test loss

If we run SGD for T gradient steps with learning rate η starting at weight θ_0, then by Taylor expansion we may express the expected test loss of the final weight θ_T in terms of statistics of the loss landscape evaluated at θ_0. As is, this Taylor series is unwieldy to write and interpret. Our technical contribution is to organize the computation of this series via combinatorial objects we call *diagrams*:

Main Idea (Informal). We may enumerate the diagrams, and we may assign to each diagram a number that depends on η, T, such that summing those numbers over all diagrams yields SGD's expected test loss. Restricting to the finitely many diagrams with $\leq d$ edges leads to $o(\eta^d)$ error. $\Diamond$

Deferring details, we illustrate the Main Idea by deriving a new result (Example 1). This shows our formalism's work flow, but only later will we explain the mathematics.

Definition 1 (Informal). Let $l_x(\theta)$ be weight θ's loss on datapoint x. We define a dictionary between (a) tensors relating to this loss landscape and (b) diagram fragments that we will soon assemble:

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

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

$$J \triangleq \mathbb{E}_x\left[\nabla\nabla\nabla l_x(\theta)\right] \triangleq \text{⋎} \qquad S \triangleq \mathbb{E}_x\left[(\nabla l_x(\theta) - G)^3\right] \triangleq \text{⸝⸝⸜}$$

Here, G, H, J denote the loss's derivatives with respect to θ, and G, C, S denote the gradient's cumulants with respect to the randomness in x. There are infinitely many analogues (with more edges), but they will not play a role in our our leading order results. Each $\nabla^d l_x$ corresponds to a degree-d node, and fuzzy outlines group nodes that occur within the same expectation.

We obtain **diagrams** by pairing together the loose ends of the above fragments. [*] For instance, we may join $C = \text{⌣⸜}$ with $H = \text{⋎}$ to get ⌢:

$$\text{⌣⸜} , \text{⋎} \quad \Longrightarrow \quad \text{●⌢●}$$
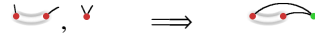
As another example, one of the ways that we may join two copies of $G = \text{➘}$ with two copies of $H = \text{⋎}$ happens to yield ⌢⌢:

$$\text{➘} , \text{⋎} , \text{⋎} , \text{➘} \quad \Longrightarrow \quad \text{●⌢●⌢●●}$$
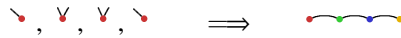
Intuitively, each diagram represents the interaction of its components: of gradients (G), noise $(C, S, \cdots)$ and curvature $(H, J, \cdots)$. In fact, §A.6 physically interprets edges as carrying information between updates and toward the test measurement. $\Diamond$

Example 1. Does non-Gaussian noise affect SGD? Specifically, let's compute how the *skewness* S affects SGD's test loss. The recipe is to identify the fewest-edged

diagrams containing $S = $ ⌣⌣. In this case, there is one fewest-edged diagram —

⌒⌒ ; it results from joining S with $J = $ ⌄. To evaluate a diagram, we multiply its components (here, S, J) with exponentiated ηH's, one for each edge (here, there are three edges). The result is easiest to write in terms of an eigenbasis of ηH:

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

This is leading order contribution of skewed noise (S) to SGD's test loss. $\diamond$

Remark 1. To understand Example 1's result, we specialize to isotropic curvature[*] and take $T \to \infty$, obtaining:

$$-\frac{\eta^3}{3!} \sum_{\mu\nu\lambda} \frac{S_{\mu\nu\lambda} J_{\mu\nu\lambda}}{3\|\eta H\|_2}$$

[*] that is, $\eta H = \|\eta H\|_2 I$

Since $J = \nabla H$, $J/\|\eta H\|_2$ measures the relative change in the curvature, H, with respect to θ. So skewed noise affects SGD in proportion to the logarithmic derivative of curvature. Gaussian approximations such as SDE miss this effect. $\diamond$

1.3 Notation and assumptions

Let $G, H, J; C, S$ be as in §1.2. They are tensors with $1, 2, 3; 2, 3$ indices, respectively. We adopt the standard sum convention: if a covector A and a vector B[†] have coefficients A_μ, B^μ, then $A_\mu B^\mu \triangleq \sum_\mu A_\mu \cdot B^\mu$. To expedite dimensional analysis, we regard the learning rate as an inverse metric $\eta^{\mu\nu}$ that converts gradient covectors to displacement vectors (see §1.1). We use the learning rate η to raise indices; thus, $H^\mu_\lambda \triangleq \sum_\nu \eta^{\mu\nu} H_{\nu\lambda}$ and $C^\mu_\mu \triangleq \sum_{\mu\nu} \eta^{\mu\nu} \cdot C_{\nu\mu}$.

[†] Vectors/covectors, also known as column/row vectors, represent distinct geometric concepts [Kolář et al., 1993].

Though η is a tensor, we may still define $o(\eta^d)$: a quantity q **vanishes to order jᵈ** when $\lim_{\eta\to 0} q/p(\eta) = 0$ for some homogeneous degree-d polynomial p.

We summarize and re-notate §1.1 as follows. We fix a loss function $l : \mathcal{H} \to \mathbb{R}$ on a space $\mathcal{H}$ of weights. We fix a distribution $\mathcal{D}$ from which unbiased estimates of l are drawn. We write l_x for a generic sample from $\mathcal{D}$ and $(l_n : 0 \le n < N)$ for a training sequence drawn i.i.d. from $\mathcal{D}$. We refer both to n and to l_n as **training points**. We assume §B.1's hypotheses, for instance that l, l_x are analytic and that all moments exist.[‡]

[‡] For example, our theory models 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.

Our general theory describes SGD with any number N of training points, T of updates, and B of points per batch. SGD then runs T many updates (i.e. $E = TB/N$ epochs, i.e. $M = T/N$ 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, $\mathcal{B}_t$, the tth batch, is sampled without replacement from the training set. For simplicity, our thesis body (but not the appendices) will assume unless otherwise stated that SGD has $E = B = 1$ and that GD has $T = B = N$.

1.4 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 Werbos [1974]'s development of back-propagation for efficient differentiation, SGD has been used to train connectionist models, for instance neural networks [Bottou, 1991], recently to remarkable success [LeCun et al., 2015].

Several lines of work treat the overfitting of SGD-trained networks. [*] 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 the aptness-to-data of not only architecture but also optimization [Neyshabur et al., 2017b]. Others approximate SGD by SDE to analyze implicit regularization, [†] 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 $T = 2$. Thus, departing from prior work, we model discrete time and hence non-Gaussian noise. Indeed, we derive corrections to continuous-time, Gaussian-noise approximations such as ordinary and stochastic differential equations (ODE, SDE). For example, we construct a loss landscape on which SGD eternally cycles counterclockwise, a phenomenon impossible with ODEs.

Our predictions are vacuous for large η. Other analyses treat large-η learning phenomenologically, whether by finding empirical correlates of generalization gap [Liao et al., 2018], by showing that **flat** minima generalize better, [‡] or by showing that **sharp** minima generalize better. [§] At least for small η, our theory reconciles these clashing claims.

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 deep 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 test 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, [¶] so we may model VAEs.

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

[†] P. Chaudhari and S. Soatto. Sgd performs variational inference, converges to limit cycles for deep networks. *ICLR*, 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; and Huan Wang, N.S. Keskar, Caiming Xiong, and R. Socher. Identifying generalization properties in neural networks. *Arxiv Preprint*, 2018

[§] C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. *Berkeley Symposium on Mathematical Probability*, 1956; Laurent Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. *ICLR*, 2017; and Lei Wu, Chao Ma, and Weinan E. How sgd selects the global minima in over-parameterized learning. *NeurIPS*, 2018

[¶] Disagreement of C and H is typical in modern learning: see Roux et al. [2012] and Kunstner et al. [2019].

2

Theory, specialized to $E = B = 1$ SGD's test loss

> Mr. Reilly was all heart. Of course, he was part valve, too.
>
> —————————————————————————
>
> John Kennedy Toole, *A Confederacy of Dunces*

A **diagram** is a finite rooted tree equipped with a partition of its nodes that obeys the *path condition*: no path from leaf to root may encounter any part more than once. We specify the root by drawing it rightmost. We draw the parts of the partition by grouping each part's nodes inside fuzzy outlines. A diagram is **irreducible** when each of its degree-2 nodes is in a part of size one.

Example 2. The diagrams ⌒, ⌢⌒ each have 2 parts; ⌢⌒, ⌢⌒ have 3. Corollaries 2, 4, 3 have $E \neq 1 \neq B$, so they feature ⌒ and ⌢, generalized diagrams that violate the path condition. Diagrams ⌒, ⌢ are irreducible; due to their green nodes, ⌢⌒, ⌢⌒ are not. For all f, $|\mathrm{Aut}_f(⌢⌒)| = 1$ and $|\mathrm{Aut}_f(⌢⌒)| = 2$. ◇

An **embedding** f of a diagram D is an injection from D's parts to (integer) times $0 \leq t \leq T$ that sends the root to T and such that, for each path from leaf to root, the corresponding sequence of times increases. So f might send ⌢⌒'s red part to $t = 3$ and its green part to $t = 4$, but — because the green node has a red child — not vice versa. Let $\left|\mathrm{Aut}_f(D)\right|$ count automorphisms of D that preserve f. And as a notational convenience, say that an edge has *duration* $|t' - t|$ under f if f maps the edge's endpoints to times t, t'. Up to unbiasing terms, [†] we construct the **re-summed value** $\mathrm{rvalue}_f(D)$ as follows:

Node rule: insert a factor a $\nabla^d l_x$ for each degree d node.

Outline rule: group each part's nodes within brackets $\mathbb{E}_x[\cdots]$.

Edge rule: insert for each duration Δt edge a factor $(I - \eta H)^{\Delta t - 1}\eta$.

We set $K \triangleq (I - \eta H)$ to write the edge rule as $K^{\Delta t - 1}\eta$. We will later interpret K as the "propagator" from past influences through time.

[†] For example, we actually define ⌣ to be the cumulant $C = \mathbb{E}\left[(\nabla l_x(\theta) - G)^2\right]$, not the moment $\mathbb{E}\left[(\nabla l_x(\theta))^2\right]$. This centering is routine (see §B.4), tedious to notate, and un-germane, so we ignore it in the thesis body.

Example 3. If f maps ⌒'s red part to time $t = T - \Delta t$, then

$$\text{rvalue}_f\left(⌒ \right) = S_{\mu\lambda\rho}(K^{\Delta t-1}\eta)^{\mu\nu}(K^{\Delta t-1}\eta)^{\lambda\sigma}(K^{\Delta t-1}\eta)^{\rho\pi}J_{\nu\sigma\pi}$$

Here, the red part gives S; the green part, J. We may integrate this expression per Remark 2 to recover Example 1. Likewise, if f maps ⌒'s red part to time $t = T - \Delta t$ and its green part to time $t = T - \Delta t'$, then

$$\text{rvalue}_f\left(⌒ \right) = G_\mu G_\lambda(K^{\Delta t-1}\eta)^{\mu\nu}(K^{\Delta t'-1}\eta)^{\lambda\sigma}H_{\nu\sigma}$$

Here, the red part gives G_μ; the green part, G_λ, and the blue part, H. ◇

2.1 Main result

Theorem 1 expresses SGD's test loss as a sum over diagrams. A diagram with d edges scales as $O(\eta^d)$, so the following is a series in η. We later truncate the series to small d, focusing on few-edged diagrams.

Theorem 1 (Special case of $E = B = 1$). *For any T: for η small enough, SGD has expected test loss*

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

Remark 2. We often content ourselves to approximate sums over embeddings by integrals over times and $(I - \eta H)^t$ by $\exp(-\eta H t)$, reducing to a routine integration of exponentials at the cost of an error factor $1 + o(\eta)$.

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

Remark 3 (Convergence 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 have not seen such pathologies in practice, so we will freely speak of "SGD in the large-T limit" as informal shorthand when referencing this Theorem.

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

For redundantly parameterized models and by empirical observation more generally, the loss landscape consists of high-dimensional "valleys" of minima.[*] It is natural to ask: *which minima within a valley does SGD prefer?* Take, for instance, the valley of Figure 2.1. As shown, a generic valley varies in width as it interpolates from sharper minima to flatter minima. That is, $J = \nabla H$ is generically non-zero.
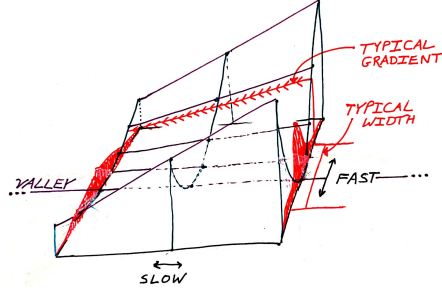
Figure 2.1: **Gradient noise pushes SGD toward flat minima.** The red densities show the typical θs, perturbed from the minimum due to noise C, in two cross sections of the loss valley. $J = \nabla H$ measures how curvature changes across the valley. Our theory does not assume separation between "fast" and "slow" modes, but we label them in the picture to ease comparison with Wei and Schwab [2019]. Compare with Figure 3.4; see Corollary 1.

Suppose now that θ_t lies directly on a minimum, and suppose that gradient noise pushes it along the "fast" direction up the valley's walls. The next gradient update will tend to pull θ_{t+1} not only back down the slope toward its original value but also in a transverse, "slow" direction toward the valley's flatter regions.

Intuitively, then, gradient noise pushes SGD toward flat minima. Theorem 1 allows us to quantify this intuition. To isolate the effect of valley width, we assume that θ is initialized exactly at a test minimum:

Corollary 1 (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*

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

We may read this result directly off of a diagram $D = $ ⌒‿•. For, 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, [*] the displacement is $\propto -C\nabla H$. That is, SGD moves toward minima that are flat *with respect to C* (Figure 2.1). Taking limits to drop the non-degeneracy hypothesis, we expect *sustained* motion toward flat regions in a valley of minima.

[*] 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.

This $T \gg 1$ result predicts a displacement of size $\Theta(\eta^2)$, while Yaida [2019b]'s similar $T = 2$ result predicts a displacement of size $\Theta(\eta^3)$. Our analysis integrates the noise over many updates, hence amplifying C's effect. Experiments verify our law.

Recalling low-pass filter theory, we recognize $CH/2 + o(C)$ as the loss increase upon convolving l with a C-shaped Gaussian; we thus say informally that SGD descends on a C-smoothed landscape. Crucially, this landscape changes as C does, and the generic behavior is that SGD's velocity field has *curl* (§3.2). We arrive at the same conclusion from a more formal perspective: while $\nabla(CH)$ is a total derivative, $C\nabla H$ is not. In short, by avoiding Wei and Schwab [2019]'s assumptions of constant C, we find that the question we posed above is ill-formed: SGD pursues directions, not locations, and it might never converge to a fixed point.

2.3 Both flat and sharp minima overfit less

It is obvious that SGD overfits less near **sharp** minima. Prior work empirically supports this claim (§1.4); after all, l_2 regularization acts by increasing the Hessian.

It is also obvious that SGD overfits less near **flat** minima. Prior work empirically supports this claim (§1.4); after all, flat minima are stable as the weight changes.

Our theory suggests a more balanced view. Per Figure 2.2, sharp minima are robust to slight changes in the average *gradient* and flat minima are robust to slight *displacements* in weight space. However, as SGD by definition equates displacements
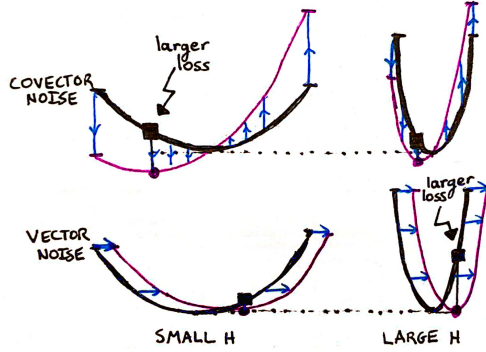

Figure 2.2: **Both curvature and the structure of noise affect overfitting.** In each of the four subplots, the ↔ axis represents weight space and the ↕ axis represents loss. ▦: *covector*-perturbed landscapes favor large Hs. ▦: *vector*-perturbed landscapes favor small Hs. SGD's implicit regularization interpolates between these rows (Corollary 2).

with gradients, it may be unclear how to reason about overfitting in the presence of curvature. Theorem 1 accounts for the implicit regularization of fixed-T descent, so it shows that both effects play a role. In fact, by routine calculus on Corollary 2, overfitting is maximized for medium minima with curvature $H \sim (\eta T)^{-1}$.

Corollary 2 (from ⌢, ⌣). *Initialize GD at a non-degenerate test minimum $\theta_\star$. The overfitting (test loss minus $l(\theta_\star)$) is*

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

and the generalization gap (test minus train loss) is:

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

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 Akaike Information Criterion (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.

Corollary 2 also recovers the Takeuchi Information Criterion (TIC): if we take $T \to \infty$, the overfitting reduces to TIC $= C/2NH$. Prior works such as Dixon and Ward [2018] uses the TIC to estimate overfitting, but they encounter numerical difficulties and require arbitrary cutoffs in the common case when H is singular. This

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

[†] **m**aximum **a** **p**osteriori, **e**vidence **lower bo**und, **m**aximum **l**ikelihood

behavior runs counter to our intuition, because effectless parameters yield singular Hs and yet should not affect overfitting. In fact, our theory recovers this intuition and shows that the implicit regularization of finite-T descent tames these singularities: the Corollary's overfitting formula tends to 0 as H shrinks.

2.4 High-C regions repel small-(E, B) SGD more than large-(E, B) SGD

Physical intuition suggests that noise repels SGD. In particular, 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! This mechanism parallels the effect of Chladni [1787] (Figure 2.3).
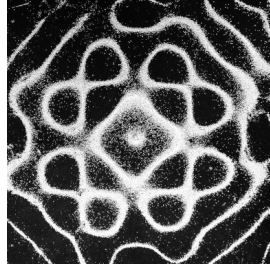
Figure 2.3: **Chladni plate**. Grains of sand on a vibrating plate tend toward stationary regions. From Pierre Dragicevic and Yvonne Jansen's data physicalization project, Creative Commons BY-SA 3.0.

More precisely, the drift is in the direction of $-\nabla C$. The effect is strongest when gradient noise is not averaged out by large batch sizes, but we may counter this effect via an artificial loss term: *

* We call the resulting optimization method **GDC**.

Corollary 3 (⌣). *SGD avoids high-C regions more than GD:* $l_C \triangleq \frac{N-1}{4N} \nabla^\mu C_\nu^\nu = \mathbb{E}\left[\theta_{GD} - \theta_{SGD}\right]^\mu - o(\eta^2)$. *If $\hat{l}_c$ is a smooth unbiased estimator of l_c, then GD on $l + \hat{l}_c$ has an expected test loss that agrees with SGD's to order η^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 and many- epoch SGD differ, to leading order, in their sensitivity to ∇C:

Corollary 4 (⌣). *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:* $\mathbb{E}\left[\theta_{M=M_0} - \theta_{M=1}\right]^\mu = \left(\frac{M_0-1}{4M_0}\right) N \left(\nabla^\mu C_\nu^\nu\right) + o(\eta^2)$.

2.5 Non-Gaussian noise affects SGD but not SDE

Landscapes with non-Gaussian are common. For example, maximum-likelihood fitting of a normal density to normally-drawn data yields a landscape whose gradients obey a χ^2 law and are, in particular, non-Gaussian (§**??**). One of our contributions is to quantify the effect of non-Gaussian noise in SGD. For example, we present

corrections to stochastic differential equations (SDE), [*] a popular theoretical approximation of SGD that assumes uncorrelated, Gaussian noise. Even if we restrict to $E = 1$ SGD to suppress correlations (§2.4), SGD and SDE differ due to non-Gaussian noise. In particular, they respond differently to changes in curvature:

Corollary 5 (⌢, ⌢⌢). *SGD's test loss exceeds ODE's and SDE's by $\frac{T}{2} C_{\mu\nu} H^{\mu\nu} + o(\eta^2)$. SGD differs from SDE due to skewed noise by $-\frac{T}{6} S_{\mu\nu\lambda} J^{\mu\nu\lambda} + o(\eta^3)$.* [†]

To further study the effect of finite T, N, we re-arrange the terms of Theorem 1: [‡]

Proposition 1. *The order η^d contribution to the expected test loss of SGD with $E = B = 1$ is:*

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

*where D has d edges. Here, D's parts have sizes $d_p : 0 \le p \le P$; $|\mathrm{ords}(D)|$ counts **total orders** of D where kids precede parents and parts are contiguous.*

[†] This approximation of Example 1's more exact expression agrees with the latter to leading order in η.

[‡] The proposition has the advantage of simplicity and the disadvantages of restrictive assumptions and weaker convergence properties: see §B.

Here, uvalues are like rvalues, but they do not depend on an embedding. [§] We organize [¶] the sum over diagrams into a double series, sorted both by their number d of edges and by their number f of fuzzy ties. [‖] Each diagram of grade (d, f) has d factors of η (by the edge rule) and f multiplications inside expectations (by the outline rule). Moreover, the multinomial coefficient is $\Theta(T^{d-f})$, since the diagram has $P = d - 1 + f$ many parts. Since $E = B = 1$, $N = T$, so:

[§] We discuss further in §B, but the details do not matter for now.

[¶] Not having established absolute convergence, we view this as a suggestive operation on formal Taylor series.

[‖] Due to our convention that we draw as few fuzzy outlines as possible, each diagram has $d - 1 - f$ parts.

$$\mathrm{uvalue}(D) \in \Theta(\eta^d T^{d-f}) = \Theta((\eta T)^d N^{-f})$$

We interpret ηT as *physical time*, since on a noiseless linear landscape, $\theta_t = \theta_0 - G\eta T$. We interpret N as the degree of *noise suppression*, since for fixed ηT SGD dynamics approaches ODE dynamics as N grows [Yaida, 2019a]. Therefore, the $(\eta T)^{\cdots} N^{-1}$ terms give the leading order corrections to continuous-time behavior. The middle

$\approx (\eta T)^3 N^{-0}$	$\approx (\eta T)^3 N^{-1}$	$\approx (\eta T)^3 N^{-2}$

three columns give the leading corrections to continuous time; they measure time-discretization. More subtly, the rightmost column gives the subleading correction, wherein non-Gaussian effects appear in the form of skewed noise ($S = $ ⌣⌣). In short, we may interpret fuzzy outlines as distinguishing SGD from ODE and SDE.

Table 2.1: **Diagrams with many fuzzy outlines scale strongly with noise suppression.** Each column shows all the total orders of some D in Proposition 1. 3-edged diagrams with f outlines contribute $\Theta((\eta N)^3 N^{-f})$ to the expected test loss. $f = 0$ depicts ODE behavior. $f = 1$ depicts the first order correction to ODE behavior. $f = 2$ depicts the second order correction to ODE behavior, with the appearance of third cumulants and hence non-Gaussian effects.

3

Experiments

> You miss 100% of the shots you don't take. — Wayne Gretzky
>
> ――――――――――――――――――――――――――――――――――――――
>
> Michael Scott, Dunder Mifflin Regional Manager

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 3.1 ▦, Chaudhari and Soatto [2018] predicts a velocity of 0 while we predict a velocity of $\eta^2/6$.

The most basic tests concerns the viability of perturbation. We directly compare 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

I bars, + signs, and shaded regions mark 95% confidence intervals based on the standard error of the mean. The label `rvalue` refers to Theorem 1's predictions, approximated as in Remark 2. Curves marked `uvalue` are polynomial approximations to Theorem 1's result (see §A.5). `uvalue`s are simpler to work with but (see ▦) may be less accurate.
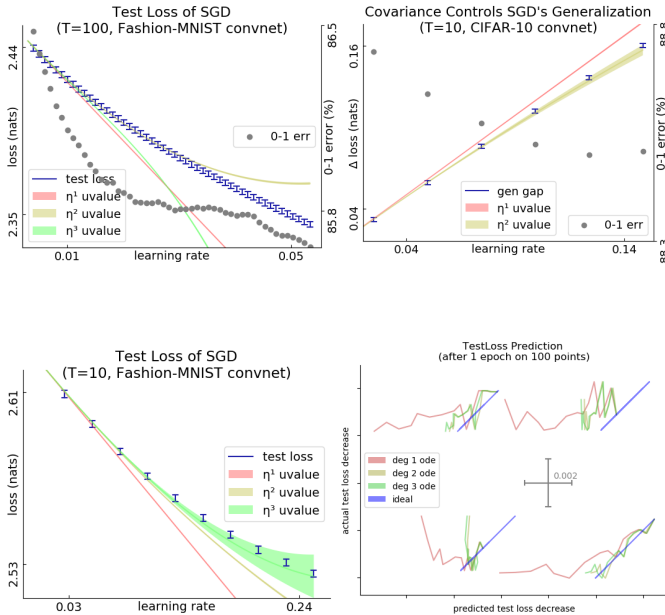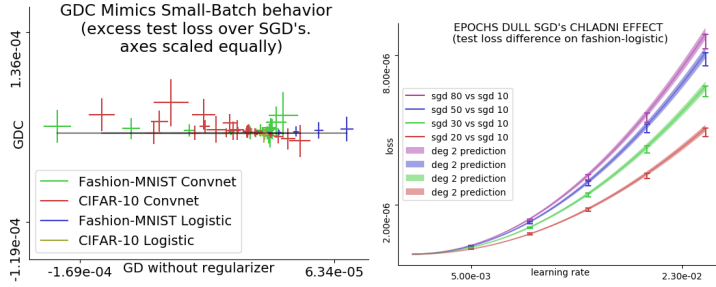

Figure 3.1: **Perturbation models SGD for small ηT.** ▪▫: Fashion-MNIST convnet's test loss vs learning rate. Here, the order-3 prediction holds until the 0-1 error improves by $5 \cdot 10^{-3}$. For large ηT, our predictions break down. ▫▪: CIFAR-10 generalization gaps. For all initializations tested (1 shown, 11 unshown), the degree-2 prediction agrees with experiment through $\eta T \approx 5 \cdot 10^{-1}$.

Figure 3.2: **Higher order terms improve order-1 predictions.** ▪▫: For all Xavier initializations Fashion-MNIST 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 $\approx 10^{-3}$. Meanwhile, order 1 predictions agree only through $\eta T \approx 10^{-1/2}$. ▫▪: We initialize CIFAR-10 near four minima found by pre-training. Order 2 predictions improve on order 1 baselines.

to increase by 0.5% (Figure 3.1). Moreover, the effort of computing higher-order terms is rewarded by increased accuracy (Figure 3.2).

3.1 Epochs and batch size; C repels SGD more than GD

Figure 3.3 fails to falsify Corollaries 3 and 4. ▫ shows that, relative to GD, high-C regions *repel* SGD. This is significant because C controls the rate at which the generalization gap (test minus train loss) grows (Figure 3.1.▫, Corollary 2). In ▫, we

Figure 3.3: C, **modulated by batch size and epoch number, controls the generalization gap.** $N = 10$ for both plots. With equal-scaled axes, ▫ shows that GDC matches SGD (small vertical variance) better than GD matches SGD (large horizontal variance) in test 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, $T = 10$. ▫: SGD with $2, 3, 5, 8$ epochs. 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.

see that multi-epoch SGD 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.

3.2 *Minima that are flat* with respect to C attract SGD

To test Corollary 1's claimed dependence on C, we construct a landscape, ARCHIMEDES, with non-constant C throughout its valley of global minima (§3.4). Figure 3.4 depicts ARCHIMEDES' chiral shape. As in Rock-Paper-Scissors, each point θ has a neighbor
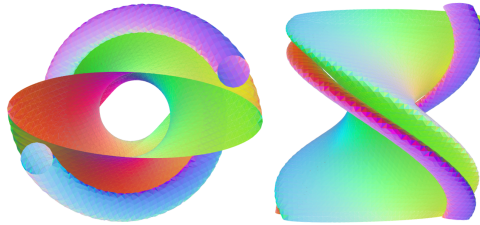

Figure 3.4: **ARCHIMEDES.** A **green** level surface of l twists around a valley of minima (z axis) at its center; l is large outside this surface. Due to anisotropic noise, θ scatters away from the z axis toward the **purple** tubes. SGD pushes the scattered θs toward lower loss, i.e. toward the level surface, and so toward larger z. The z axis points into the page (▪▫) or upward (▫▪). We made these plots with the help of Paul Seeburger's online applet, CalcPlot3D.

that, from $C(\theta)$'s perspective but not absolutely, is flatter. This permits eternal motion despite the landscape's symmetry. Indeed, Corollary 1 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. [*] Our prediction agrees with experiment (Figure 3.5.▫). Note that 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.

Because SGD's motion depends smoothly on the landscape, the special case of ARCHIMEDES implies that curl is typical. 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 curl in SGD's velocity shows that no such $\tilde{l}$ exists.

Figure 3.5.▫ further compares SDE to SGD. Two effects not modeled by SDE — time-discretization and non-Gaussian noise oppose on this landscape but do not

[*] Indeed, ARCHIMEDES' velocity is η-perpendicular to the image of $(\eta C)^\mu_\nu$ in tangent space.
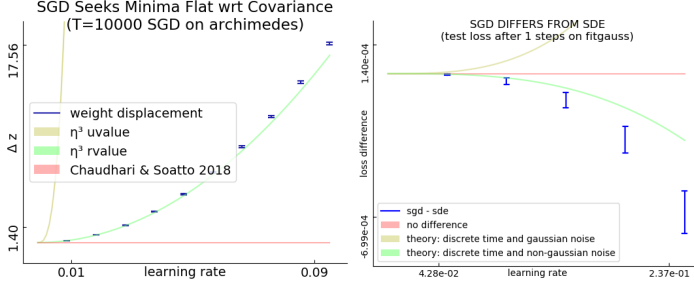
Figure 3.5: **Predictions near minima excel for large** ηT**.** ▪▫: SGD travels Archimedes' valley of global minima in the positive z direction. ▫▪: SGD's difference from SDE after $\eta T \approx 10^{-1}$ with maximal coarseness on Gauss.

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.

3.3 *Sharp and flat minima both overfit less than medium minima*

We test Corollary 2's predictions about generalization and minima. The balance between flat and sharp minima is non-trivial even in Mean Estimation (§3.4): Near $H = 0$, our predictions improve on TIC [Dixon and Ward, 2018] and thus on AIC (Figure 3.6.▪▫).
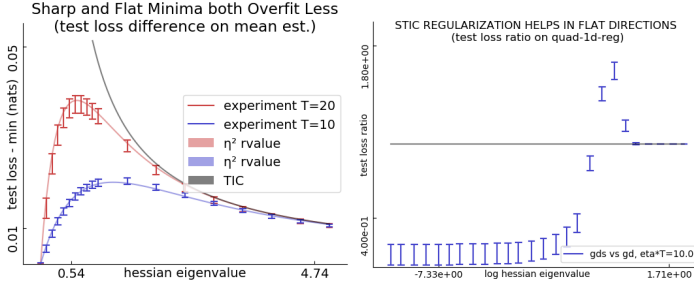

Figure 3.6: **Both sharp and flat minima overfit less.** ▪▫: For Mean Estimation with fixed C and a range of Hs, initialized at the truth, the test losses after fixed-T GD are smallest for very sharp and very flat H. ▫▪: Blue intervals regularization using Corollary 2. When the blue intervals fall below the black bar, this proposed method outperforms plain GD.

To combat overfitting, we may add Corollary 2's expression for generalization gap to l. By descending on this regularized loss, we may tune smooth hyperparameters such as l_2 regularization coefficients for small datasets ($H \ll C/N$) (Figure 3.6.▫▪). 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. Since matrix exponentiation takes time super-quadratic in dimension, this regularizer is most useful for small models.

3.4 Artificial Landscapes

We defined three artificial landscapes: GAUSS, ARCHIMEDES, and MEAN ESTIMATION.

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. [*] 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 h 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 test loss increases. At $h = 0$, the expected gradient vanishes, and the test loss of SGD involves only diagrams that have no leaves of size one.

ARCHIMEDES

The ARCHIMEDES landscape has chirality, much like its namesake's screw. [†] Specifically, the ARCHIMEDES 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 thesis body showed that SGD travels in ARCHIMEDES' $+z$ 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.

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 **??** ▫▪) 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 $\lambda = 0$.

3.5 Image-classification landscapes

Architectures

In addition to the artificial loss landscapes Gauss, Archimedes, 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 [*] and Fashion-MNIST [†] datasets. 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.

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

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

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 test 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 test and training sets to overlap, hence biasing test loss measurements. In fact, test 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, for example 10^1 datapoints out of $\sim 10^{4.5}$ dirac masses, so overlaps are rare.

3.6 Measurement process

Diagram evaluation on real landscapes

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

Descent simulations

We recorded test and train 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 ARCHIMEDES with SGD with $T = 10^4$ and η ranging from 10^{-2} to 10^{-1}. We ran 10^3 trials of MEAN ESTIMATION with 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.

Implementing optimizers

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

Our GDC regularizer was implemented using the unbiased estimator

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

For our tests of regularization based on Corollary 2, 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.

4
Conclusion

> If you will recollect, we are now in Autumn —
> season of mists and mellow fruitfulness.
> ———————————————————
> Reginald Jeeves of the Berkeley Mansions

4.1   Contributions

We presented a diagram-based method for studying stochastic optimization on short timescales or near minima. Corollaries 1 and 2 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 conjectured by Roberts [2018]. This is significant because, while small batch sizes can lead to better generalization, [†] modern infrastructure increasingly rewards large batch sizes. [‡]

Machine learning today has an enormous carbon footprint. [§] Our analysis of SGD may lead to a reduced carbon footprint in two ways. **First**, §**??** shows how to modify the loss landscape so that large-batch GD enjoys the stochastic regularizing properties of small-batch SGD, or (symmetrically) so that small-batch SGD enjoys the stability of large-batch GD. By unchaining the effective batch size from the actual batch size, we raise the possibility of training neural networks on a wider range of hardware than currently practical. For example, asynchronous concurrent SGD [¶] might require less inter-device communication and therefore less power. **Second**, the modification of AIC developed in §2.3 and §3.3 permits certain forms of model

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

[‡] 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. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in nlp. *ACL*, 2019

[¶] Feng Niu, B. Recht, C. Ré, and S.J. Wright. Hogwild!: A lock-free approach to parallelizing sgd. *NeurIPS*, 2011

selection by gradient descent rather than brute force search. More generally, our analysis may inform the design of meta-learning outer loops such as MAML. [*] This might drastically reduce the energy consumed during model selection.

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

Since our predictions depend only on loss data near initialization, they break down after the weight moves far from initialization. Our theory thus best applies to small-movement contexts, whether for long times (large ηT) near an isolated minimum or for short times (small ηT) in general. Thus, the theory might help to analyze meta-learners based on fine-tuning (e.g. [Finn et al., 2017]'s MAML).

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

4.2   Future topics

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

[†] Landau and Lifshitz [1960, 1951] review these concepts.

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

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

Our work assumes a flat metric $\eta^{\mu\nu}$, but it might generalize to weight spaces curved in the sense of Riemann. [‡] Such curvature finds concrete application in the *learning on manifolds* paradigm, [§] 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:

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

[§] P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization algorithms on matrix manifolds, chapter 4. *Princeton University Press*, 2007; and Hongyi Zhang, S.J. Reddi, and S. Sra. Fast stochastic optimization on riemannian manifolds. *NeurIPS*, 2016

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 generalization 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.

Bibliography

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

Y.S. Abu-Mostafa, M. Magdon-Ismail, and Hsuan-Tien Lin. Learning from data. *Caltech*, 2012.

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.N. Bernstein. Sobranie sochinenii. *Moscow*, 1964.

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.

P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and E. Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *ICLR*, 2017.

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

V. Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 1979.

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

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

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

G. Gur-Ari, D.A. Roberts, and E. Dyer. Gradient descent happens in a tiny subspace. *ArXiV preprint*, 2019.

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

R. Impagliazzo and V. Kabanets. Constructive proofs of concentration bounds. *Chapter LNCS volume 6302)*, 2010.

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.

W. Mulzer. Five proofs of chernoff's bound with applications. *Bulletin of the European Association for Theoretical Computer Science*, 2018.

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.

Feng Niu, B. Recht, C. Ré, and S.J. Wright. Hogwild!: A lock-free approach to parallelizing sgd. *NeurIPS*, 2011.

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.

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.

T. Steinke and J. Ullman. Subgaussian tail bounds via stability arguments. *ArXiV preprint*, 2017.

E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in nlp. *ACL*, 2019.

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

Appendices A and B serve respectively as informal and formal guides to our theory. The first is a user-friendly tutorial; the second contains our proofs. Appendix C describes some results only loosely related to our main work.

A

How to calculate test losses

> Talking about pumpkins doesn't make them grow.
>
> ―――――――――――――――――
>
> Mma Ramotswe of the No. 1 Ladies' Detective Agency

Our work introduces a novel technique for calculating the expected learning curves of SGD in terms of statistics of the loss landscape near initialization. Here, we explain this technique. New combinatorial objects — *space-time grids* — arise as we relax the thesis body's assumption that $E = B = 1$. This, too, we will explain. We note for now that there are are **four steps** to computing the expected test loss, or other quantities of interest, after a specific number of gradient updates:

- **Draw, as a space-time grid**, the batches, training set size, and number of epochs.

- **Draw embeddings**, of relevant diagrams into the space-time grid.

- **Evaluate each diagram embedding**, whether exactly (via what we will call rvalues) or roughly (via what we will call 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 test losses, we describe how the four steps also give us variances, train losses, of weight displacements.

A.1 *An example calculation: the effect of epochs*

Question 2. *How does multi-epoch SGD differ from single-epoch SGD? What is the difference between the expected test 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 test losses, namely $l(\theta_0) - Tl_\mu \eta^{\mu\nu}$. So Question 2's answer will be second-order (or higher-order) in η.

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 2: one for multi-epoch SGD and another for single-epoch SGD — see Figure A.1.
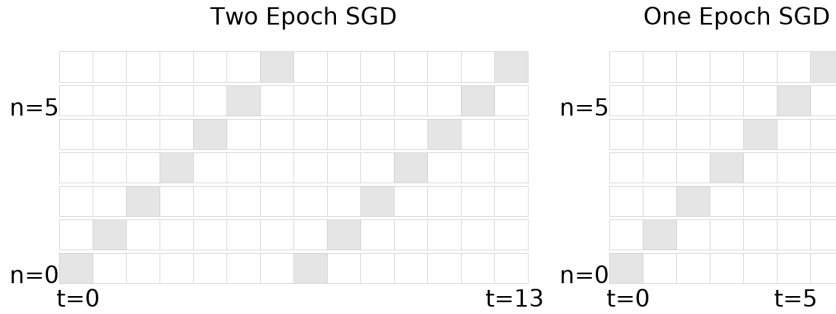
Figure A.1: **The space-time grids of single-epoch and of multi-epoch SGD.** A cell at row n and column t is shaded provided that the nth training sample inhabits the tth batch. Both grids depict $N = 7$ training points and batch size $B = 1$; neither depicts training-set permutation between epochs.
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.

Embeddings of diagrams into space-time

There are four two-edged diagrams: ⌢, ⌢, ⌢, and ⌢. We permit the diagram ⌢, which violates the path condition mentioned in §2, 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 A.2 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 2.
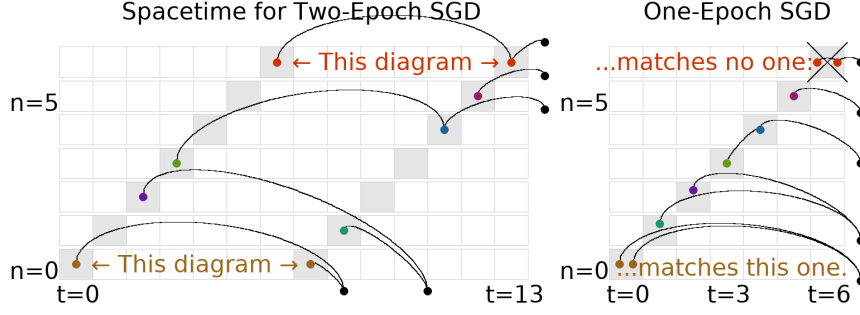
Figure A.2: **The diagram** ⌢ **embeds into multi-epoch but not single-epoch space-time.** Drawn on each of the two grids are examples of embeddings. The black nodes external to the grids are positioned arbitrarily. From top to bottom in each grid, the five diagrams embedded are ⌢ (or ⌢), ⌢, ⌢, ⌢, and ⌢ (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.

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 test loss, we will we will multiply each diagram's uvalue by the number of ways that diagram embeds.

Figure A.2 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 A.2 shows that this cancellation happens for all of the order-2 diagrams *except* for ⌢. Therefore, to second order, the answer to Question 2 will be (some multiple of) uvalue(⌢).

To compute ⌢'s value, we follow the rules in Section 2; the edge rule for uvalues is that each edge becomes an η. So

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

Sum of the values

Referring again to Figure A.2, 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 test 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 test loss of $M = M_0$ SGD exceeds the test loss of $M = 1$ SGD by this much:

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

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

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

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

Remark. An essentially similar argument proves Corollary 4.   ◇

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 A.3.
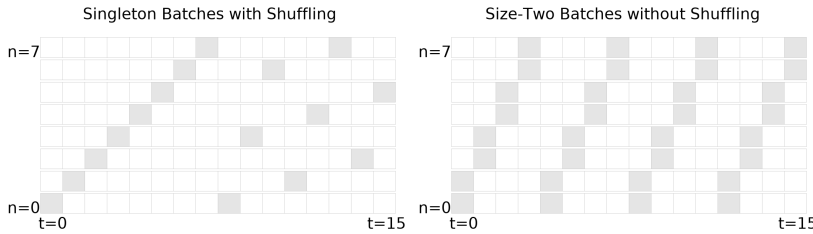
Figure A.3: **The space-time grids of two SGD variants.** Shaded cells show (n, t) pairs (see text).
Left: Two epoch SGD with batch size one. The training set is permuted between epochs.
Right: Four epoch SGD with batch size two. The training set is not permuted between epochs.

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. [*]   ◇

[*] A routine check shows that for fixed T, inter-epoch shuffling yields only an $o(\eta^3)$ effect on test 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 thesis body; in particular, we no longer require the thesis 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 A.4.

Figure A.4: 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.
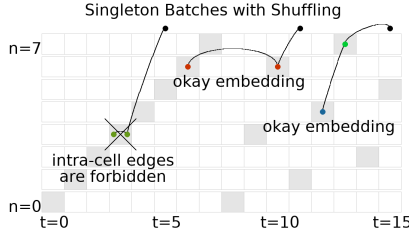
In principle, the relevant diagrams for a calculation with error $o(\eta^d)$ are the diagrams with at most d edges. For d greater than 2, there will be many such diagrams. However, in practice we gain insight even from considering one diagram at a time:

Remark. In this thesis's corollaries, we seek to extract the specific effect of a specific landscape or optimization feature such as skewed noise (Example 1) or multiple epochs (§A.1). In these cases, it is usually the case that most diagrams are irrelevant. For example, because a diagram evaluates to a product of its components, the only way the skewness of gradient noise can appear in our calculations is through diagrams such as ⌢ that have a part of size 3. Thus, the analysis in Example 1 was able

to ignore diagrams such as . Likewise, in §A.1 we argued by considering which embeddings that the only diagram relevant to Question 2 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 A.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.

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 thesis 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 $T \to \infty$.

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):

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

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

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

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

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

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

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

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

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

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

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
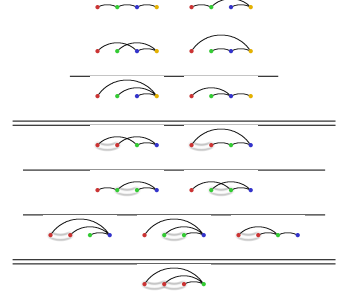
Table A.1: **Multiple ways to draw the 6 distinct degree-3 diagrams for $B = E = 1$ SGD's test loss.** Because the space-time of $B = E = 1$ SGD has only one cell per row and one cell per column, the only diagrams that have a non-zero number of embeddings are the diagrams that obey §2's path condition. 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.

expectation brackets $\mathbb{E}_x\left[\cdot\right]$ — 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, 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 4. 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 5 (Un-resummed value). Remember that $\overset{\smile}{} = C_{\mu\nu}$ and $\vee = H_{\lambda\rho}$, so that $\overset{\smile}{}\vee = C_{\mu\nu}H_{\lambda\rho}$. Then

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

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

Resummed values: $\mathrm{rvalue}_f(D)$

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 6 (Re-summed value). Recall as in Example 5 that $\overset{\smile}{} = C_{\mu\nu}$ and $\vee = H_{\lambda\rho}$, so that $\overset{\smile}{}\vee = C_{\mu\nu}H_{\lambda\rho}$. Then if f is an embedding of $\overset{\frown}{}$ that sends the diagram's red part to a time t (and its green root to T), we have:

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

Here, $\overset{\frown}{}$ 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. ◇

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 thesis body generalizes to

Theorem. *For any T: for η small enough, SGD has expected test 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 7. Let us return to $D = $ ⌢ , embedded, say, in the space-time of one-epoch one-sample-per-batch SGD. From Example 6, we know that we want to sum the following value over all embeddings f, i.e. over all $0 \leq t < T$ to which the red part of the diagram's partition may be assigned:

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

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

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

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

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

and $A = (\eta H) \otimes I + I \otimes (\eta H)$, we evaluate the integral as:

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

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

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

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

In short, we sum embeddings of uvalues directly. We sum embeddings of rvalues using an integral-of-exponentials approximation along with the rule $\int_{0 \le u < T} du \, \exp(-uA) = (I - \exp(-TA))/A$. 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 1.

A.6 Interpreting diagrams intuitively

We may intuitively interpret edges as carrying influence from the training set toward the test measurement. See Figure A.5. From this perspective, we may intuitively
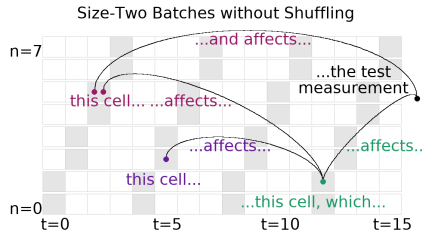
Figure A.5: **Edges carry information**. Embedding of a 4-edged diagram.

interpret edges in an rvalue calculation as carrying influence from the training set toward the test measurement. As shown in Figure A.6, each resummed value
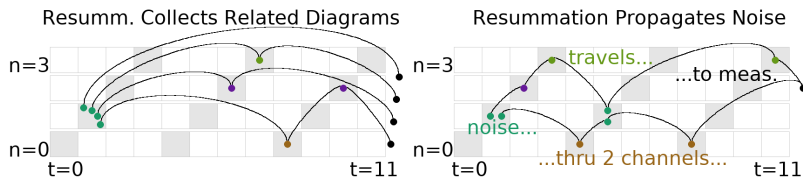
Figure A.6: **Resummation propagates information, damped by curvature**. **Left**: Here is one of many unresummed terms captured by a single resummed embedding for ⌢. **Right**: each resummed value represents many un-resummed values. Here is one of many un-resummed terms captured by a single resummed embedding for ⌢.

represents many un-resummed values, each modulated by the Hessian (ᵛ) in a different way.

A.7   How to solve variant problems

In §4.2, we briefly discuss second-order methods and natural gradient descent. Here, we briefly discuss modifications. We omit proofs, which would closely follow §B's proof of the expectation-of-test-loss case.

Variance (instead of expectation)

To compute variances instead of expectations (with respect to the noise in the training set), one considers generalized diagrams that have "two roots" instead of one. More precisely, to compute, say, the un-centered second moment of test 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 ⬭ or

⬭⬭ may appear in this computation.

Measuring on the training (instead of test) set

To compute the training loss, we compute with all the same diagrams as the test 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. test loss minus training loss), we sum over all the diagrams that expressly violate this constraint (and then, since generalization gp is test minus train instead of train minus test, we multiply the whole answer by -1). For example, the generalized diagrams ⬭ or ⬭⬭ may appear in this computation.

Weight displacement (instead of loss)

To compute displacements instead of losses, one considers generalized diagrams that have a "loose end" instead of a root. For example, the generalized diagrams ⌄ or

⬭ may appear in this computation.

A.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 [*] 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

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

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 .

Effect of batch size

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

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 such embeddings in GD's spacetime, so GD's test 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:

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

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:

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

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:

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

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

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

Likewise,

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

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

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

and

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

$\cdots$ 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 test 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

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

Effect of non-Gaussian noise at a minimum.

We consider vanilla SGD initialized at a local minimum of the test loss. One expects θ to diffuse around that minimum according to gradient noise. We compute the effect on test loss of non-Gaussian diffusion. Specifically, we compare SGD test loss on the loss landscape to SGD test 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.

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

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

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:

$$\theta_{t+1} = \theta_t - \eta \nabla l_{n_t}(\theta_t)$$

$$\in \theta_0 + \eta a_t + \eta^2 b_t + \eta^3 c_t$$

$$- \eta \left(\nabla l_{n_t} + \nabla^2 l_{n_t}(\eta a_t + \eta^2 b_t) + \frac{1}{2} \nabla^3 l_{n_t}(\eta a_t)^2 \right) + o(\eta^3)$$

$$= \theta_0 + \eta \left(a_t - \nabla l_{n_t} \right)$$

$$+ \eta^2 \left(b_t - \nabla^2 l_{n_t} a_t \right)$$

$$+ \eta^3 \left(c_t - \nabla^2 l_{n_t} b_t - \frac{1}{2} \nabla^3 l_{n_t} a_t^2 \right) + o(\eta^3)$$

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 test loss of SGD is

$$l(\theta_T) \in l(\theta_0) + (\nabla l)(\eta a_T + \eta^2 b_T + \eta^3 c_T)$$

$$+ \frac{\nabla^2 l}{2}(\eta a_T + \eta^2 b_T)^2$$

$$+ \frac{\nabla^3 l}{6}(\eta a_T)^3 + o(\eta)^3$$

$$= l(\theta_0) + \eta\left((\nabla l) a_T \right)$$

$$+ \eta^2 \left((\nabla l) b_T + \frac{\nabla^2 l}{2} a_T^2 \right)$$

$$+ \eta^3 \left((\nabla l) c_T + (\nabla^2 l) a_T b_T + \frac{\nabla^3 l}{6} a_T^3 \right) + o(\eta)^3$$

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:

$$\mathbb{E}[a_T b_T] = -\sum_t \sum_{t_0 < t_1} \mathbb{E}\left[\nabla l_{n_t} \nabla^2 l_{n_{t_1}} \nabla l_{n_{t_0}} \right]$$

$$= -\sum_{t_0 < t_1} \sum_{t \notin \{t_0, t_1\}} \mathbb{E}[\nabla l_{n_t}] \mathbb{E}\left[\nabla^2 l_{n_{t_1}} \right] \mathbb{E}\left[\nabla l_{n_{t_0}} \right]$$

$$- \sum_{t_0 < t_1} \sum_{t = t_0} \mathbb{E}\left[\nabla l_{n_t} \nabla l_{n_{t_0}} \right] \mathbb{E}\left[\nabla^2 l_{n_{t_1}} \right]$$

$$- \sum_{t_0 < t_1} \sum_{t = t_1} \mathbb{E}\left[\nabla l_{n_t} \nabla^2 l_{n_{t_1}} \right] \mathbb{E}\left[\nabla l_{n_{t_0}} \right]$$

Since $\mathbb{E}\left[\nabla l\right]$ divides $\mathbb{E}\left[a_T b_T\right]$, 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 test loss, we get

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

B

Mathematics of the theory

B.1 Assumptions and Definitions
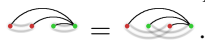
We assume throughout the following regularity properties of the loss landscape.

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.

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 $\mathbb{E}$ and ∇ 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. [†] A metric-independent way of expressing this boundedness constraint is that the gradients all lie in some subset $\mathcal{S} \subseteq T^\star \mathcal{H}$ of the cotangent bundle of weight space, where, for any compact $C \subseteq \mathcal{H}$, we have that the topological pullback — of $\mathcal{S} \hookrightarrow T^\star \mathcal{H} \twoheadrightarrow \mathcal{H}$ and $C \hookrightarrow \mathcal{H}$ — 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

[†] 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.

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 uvalue(D) and 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]'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:*

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

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 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

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

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

Condition B.2 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*

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

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:

$$\cdots = \sum_{0 \le d < \infty} (-\eta)^d \sum_{\substack{(d_t : 0 \le t < T) \\ \sum_t d_t = d}} \left(\prod_{0 \le t < T} \frac{(g\nabla)^{d_t}}{d_t!}\bigg|_{g = \nabla l_t(\theta)}\right) s(\theta) \qquad \text{(B.3)}$$

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

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

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 B.3 (that features c_d appear as coefficients) converges to an analytic function for all η bounded as in condition B.2. 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. □

B.3   From Dyson to diagrams

We now describe the terms that appear in the Key Lemma. The following result looks like Theorem 1, except it has $\text{uvalue}(D)$ instead of $\text{rvalue}_f(D)$, 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*

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

Here, D is a diagram whose root r does not participate in any fuzzy edge, f is an embedding of D into spacetime, and $\left|\mathrm{Aut}_f(D)\right|$ counts the graph-automorphisms of D that preserve f's assignment of nodes to cells.

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

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

$$\sum_{\substack{D\in\mathrm{im}(\mathcal{F}) \\ \text{with } d \text{ edges}}} \left(\sum_{f:D\to\mathcal{F}(S)} \frac{1}{\left|\mathrm{Aut}_f(D)\right|} \right) \frac{\mathrm{uvalue}_\pi(D,f)}{B^d} \tag{B.4}$$

Here, uvalue_π is the value of a diagram embedding before taking expectations over train sets. We have for all f that $\mathbb{E}\left[\mathrm{uvalue}_\pi(D,f)\right] = \mathrm{uvalue}(D)$. Observe that both sides of B.4 are finitary sums.

Remark 4 (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 terms indexed by functions v 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 $\mathcal{H} = \min S \subseteq S$ 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 B.4. 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, $g = \sum_{m \in \mathcal{H}} \nabla l_m(\theta)/B$ expands to $B^{d_0}(d - d_0 + 1)^{d_0}$ terms, each conveniently indexed by a pair of functions $\beta : [d_0] \to \mathcal{H}$ 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 4, 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

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

by the Orbit Stabilizer Theorem of elementary group theory.

It is thus enough to show that

$$\left| \text{Aut}_f(D) \right| = \left| \text{Aut}_{\tilde{f}}(D) \right| \prod_{(m,i) \in \mathcal{H} \times \tilde{D}} \left| (\beta \times v)^{-1}(m, i) \right|! \tag{B.5}$$

We will show this by a direct bijection. First, observe that $f = \beta \sqcup \tilde{f} : [d_0] \sqcup \tilde{D} \to \mathcal{H} \sqcup \tilde{S}$. 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 $\prod_{m \in \mathcal{H}} |(\beta \times \nu)^{-1}(m, i)|!$ many choices for σ_i. Claim B.5 follows, and with it the correctness of coefficients. $\qquad \square$

Remark 5 (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 1. That proposition immediately follows from the now-proven Theorem 3.

B.4 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.

Remark 6. We say an embedding is **strict** if it assigns to each part a different datapoint n. Then, by Möbius inversion, [*] a sum over strict embeddings of moment values (§A.4) matches a sum over all embeddings of uvalues. $\qquad \diamond$

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

The difference in loss from the noiseless case is then 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

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

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

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

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

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

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

Meanwhile, the orbit is all of $|X|$. Indeed, suppose $i_A, i_B \in X$. We will present $\psi \in \mathrm{Aut}_f(\rho)$ such that $i_B \sim i_A \circ \psi$ by $\phi = $ 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. $\square$

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

## B.5 Proofs of corollaries

Corollary 1

Proof. The relevant irreducible diagram is colorred (amputated as in the previous subsection). 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:

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

Standard calculus then gives the desired result. $\qquad\square$

Corollary 2's first part

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:

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

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. $\qquad\square$

Corollary 2's second part

We apply the generalization gap modification (described in §A.7) to Theorem 1's result about test losses.

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,

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

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

Corollaries 4 and 3

Corollary 4 and Corollary 3 follow from plugging appropriate values of M, N, B into the following proposition.

Proposition 2. *To order η^2, the test loss of SGD — on N samples for M epochs with batch size B dividing N and with any shuffling scheme — has expectation*

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

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

diagram	embed.s w/ $\left|\text{Aut}_f\right| = 1$	embed.s w/ $\left|\text{Aut}_f\right| = 2$
•	1	0
⌒	MNB	0
⌒⌒	$\binom{MNB}{2}$	0
⬭	$N\binom{MB}{2}$	0
⌒	$\binom{MNB}{2}$	0
⬭	$N\binom{MB}{2}$	MNB

Proof of Proposition 2. To prove Proposition 2, 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 B.5. □

Corollary 5

The corollary's first part follows immediately from Proposition 2.

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.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 $\prod_{0\le d<D} X_d$, 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 $\mathsf{A}_{0\le s<2S}\, x_0^{(s)} x_1^{(s)}$, where A denotes averaging over the sample. In fact, the following also works:

$$S\left(\underset{0\le s<S}{\mathsf{A}}\, x_0^{(s)}\right)\left(\underset{0\le s<S}{\mathsf{A}}\, x_1^{(s)}\right) + (1 - S)\left(\underset{0\le s<S}{\mathsf{A}}\, x_0^{(s)}\right)\left(\underset{S\le s<2S}{\mathsf{A}}\, x_1^{(s)}\right) \quad\quad \text{(B.7)}$$

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

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

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

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

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

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

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

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

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

Solving B.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

Bonus tracks

The lost glove is happy.

———————————————

Zemblan proverb

Our study of SGD and generalization theory led to a few miscellaneous ideas separate from the main results we report above. We briefly describe them.

C.1 Long-term prediction of SGD dynamics is intractable

The proposition is not surprising, but the lemma by which we prove it might be.

Proposition 3. *If there exists a polynomial time algorithm for computing the ith bit of the expected test loss after T steps of SGD on a given landscape,[†] then $P = NP = PSPACE$. Here, the landscape is given as an oracle for the jth bit — assumed well-defined — of any well-formed expression involving expectations, products, and derivatives applied to the landscape.*

[†] To keep the theorem non-trivial, we allow only landscapes expressible in terms of elementary functions; for example, landscapes that evaluate to a non-computable constant are disallowed.

This proposition situates the short-term results of our thesis by justifying the meteorological analogy of §4: though our short-term predictions extend to long-term qualitative insight, we do not expect any quantitative long-term theory at the levels of precision of our short-term theory.

First we show that there is a loss landscape on which SGD eternally cycles.

Lemma 1 (Ratchet Landscape)**.** *There is a loss landscape (whose weight space is the circle) on which, for fixed η sufficiently small, SGD's net displacement after T steps of is bounded below with probability 1 and bounded below by some fixed strictly increasing affine function with probability $1 - \exp(-T)$.*

Proof of lemma. Let

$$\mathrm{bump}(\theta; a, b) = \begin{cases} \exp\left(-\frac{1}{(\theta-a)(b-\theta)}\right) & \theta \in (a, b) \\ 0 & \text{else} \end{cases}$$

be a smooth bump function with support $[a, b]$. Define a loss landscape on the circle

S^1 (identified with $\mathbb{R}/2\pi\mathbb{Z}$ with representatives $[-\pi,\pi)$) as follows:

$$l_x(\theta) = \epsilon \cdot \sin(\theta) + x \cdot \theta \cdot \text{bump}(\theta; -\pi/2 - \epsilon, \pi/2 + \epsilon)$$

where $x \sim \text{Uniform}(-1, +1)$ and $0 < \epsilon < (\pi - 2)/2 \wedge 1$.

Now consider SGD with $4\eta < 1/\pi$. By construction, SGD will displace the weight by < 1 in either direction, so the intervals $[\theta_t, \theta_{t+1}]$ will always be contained fully in $I = (-\pi/2 - \epsilon, \pi/2 + \epsilon)$, fully in $S^1 \setminus I$, or will cross the boundary of I exactly once. When $\theta \notin I$, the updates will be noiseless and therefore θ will deterministically increase until it enters I. When $\theta \in I$, then θ will undergo a random walk until it exits I. The steps of this walk may take either sign, so with positive probability, the net displacement during this walk is positive. Therefore, noting that the tail masses of the distribution of the displacement after some large, fixed number of steps are bounded above by those of some Bernoulli, we finish by applying Chernoff's bound. □

Remark 7. The lemma may look similar to the behavior of ARCHIMEDES, described in the thesis body. However, ARCHIMEDES had better average-case behavior and worse worst-case behavior. The typical period of the Rachet lemma's cycling decays exponentially with $1/\eta$, while the typical period for ARCHIMEDES decays as an inverse-square power law. That said, ARCHIMEDES can go arbitrarily far backward, albeit with tiny probability, while the Rachet lemma's construction cannot. ◇

Sketch of Proposition 3's proof. We show how to simulate a PSPACE Turing machine $\mathcal{M}$ on a given length-n input in only polynomial time by constructing an appropriate loss landscape. The idea is to encode the machine's state as a point in the weight space $\mathcal{M}$. In particular, if p is a polynomial for which at most $p(n)$ cells of the tape will be used, and if $\mathcal{M}$ has q many head states, then we let $\mathcal{H} = \mathbb{R}^{p(n)+p(n)+q} \times S^1$. We consider a point

$$(c, x, h, t) \in \{0, 3\}^{p(n)} \times \{0, 3\}^{p(n)} \times \{0, 3\}^q \times S^1 \subseteq \mathcal{H}$$

to represent a tape state c, a head location x (by one-hot encoding), a head state h (by one-hot encoding), and an auxilliary phase t that we will call the *clock*.

For each of the polynomially many $(2 \cdot q \cdot p(n))$ possible tuples of cell-value-at-head, head state, and head position, we add a term to the loss landscape whose gradients lie completely within the subspace of dimension $3 + 3 + q + 1$ spanned by three tape-state axes at and neighboring the head, the three head-location axes at and neighboring the head, al the head-state axes, and the clock axis. The term is masked by a $(3 + 3 + q + 1)$-dimensional bump function so that when and only when we are nearby a point representing the input tuple do we move, completing one cycle around the clock, until we are close to the point representing the next tuple according to $\mathcal{M}$'s definition. We use the Rachet lemma to shape this term so that the simulation of $\mathcal{M}$ goes forward in time.

By adding special bump functions with large and distinct values for accept and reject states, we may query via the Ratchet lemma for whether the accept or reject state was reached after $2^p(n) \times p(n) \times q$ steps. □

C.2 A new proof of a Chernoff bound

Suppose we flip a biased coin N times. Intuitively, the resulting fraction of flips that yield heads usually does not far exceed its expectation p. More formally:

Theorem 4 (Chernoff). *Let x be the average of N many i.i.d. Bernoullis with parameter p. For $0 < g$, x exceeds $p + g$ with probability at most $\exp(-Ng^2)$.*

We understand Chernoff's bound as quantifying how fast the Central Limit Theorem kicks in. With x, p representing the training and test errors of a specific hypothesis, the theorem controls the generalization gap. Due to this application, the bound and its variants are fundamental to the statistical theory of generalization in learning. It is thus interesting to understand Chernoff's bound in multiple ways. For example, Mulzer [2018] surveys five distinct proofs, which respectively employ moment generating functions, [*] the binomial theorem, [†] direct product theorems, [‡] weight functions, [§] and differential privacy. [¶] We now present what we believe to be an original and particularly conceptual proof.

[*] S.N. Bernstein. Sobranie sochinenii. *Moscow*, 1964
[†] V. Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 1979
[‡] R. Impagliazzo and V. Kabanets. Constructive proofs of concentration bounds. *Chapter LNCS volume 6302)*, 2010
[§] W. Mulzer. Five proofs of chernoff's bound with applications. *Bulletin of the European Association for Theoretical Computer Science*, 2018
[¶] T. Steinke and J. Ullman. Subgaussian tail bounds via stability arguments. *ArXiV preprint*, 2017

Proof. We'll switch viewpoints: flipping a coin is like choosing a boxed point on a stick where green means tails and red means heads.
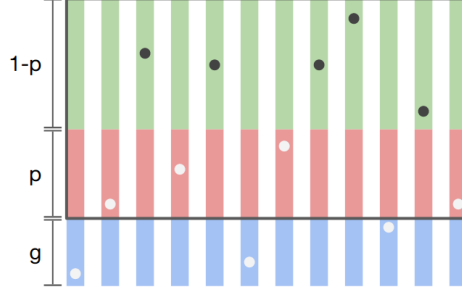

Figure C.1: *We randomly select points on N vertical sticks. Each stick has three parts:* **green** *with length $1 - p$,* **red** *with length p, and* **blue** *with length g. We call non-blue points* **boxed** *and non-green points* **hollow**. *Shown are 9 boxed points and 7 hollow ones.*

We'll bound the chance that at least $M_0 = (p + g)N$ heads appear. That is, we will bound the conditional probability — given that all points are boxed — that at least M_0 points are red. For any $M \geq M_0$:

$\mathbb{P}[M \text{ are red} \mid \text{all are boxed}]$

$= \mathbb{P}[M \text{ are red} \wedge \text{all are boxed}] \, / \, \mathbb{P}[\text{all are boxed}]$

$= \mathbb{P}[M \text{ are hollow} \wedge \text{all hollows are red}] \, / \, \mathbb{P}[\text{all are boxed}]$

$= \mathbb{P}[M \text{ are hollow}] \cdot \mathbb{P}[\text{all hollows are red} \mid M \text{ are hollow}] \, / \, \mathbb{P}[\text{all are boxed}]$

$= \mathbb{P}[M \text{ are hollow}] \cdot (1 + g/p)^{-M} \, / \, (1 + g)^{-N}$

$\leq \mathbb{P}[M \text{ are hollow}] \cdot (1 + g/p)^{-M_0} \, / \, (1 + g)^{-N}$

Since the above holds for all $M \geq M_0$, we conclude:

$$\mathbb{P}[\text{at least } M_0 \text{ are red} \mid \text{all are boxed}]$$

$\leq (1 + g/p)^{-M_0}/(1+g)^{-N}$ $\qquad$ probabilities are at most 1

$\leq \exp(-M_0 g/p) \exp(Ng)$ $\qquad$ exp is convex

$= \exp(-(p+g)Ng/p + Ng) = \exp(-Ng^2/p)$ $\qquad$ $M_0 = (p+g)N$

$\leq \exp(-Ng^2)$ $\qquad$ probabilities are at most 1

This is **Chernoff's bound** for coin flips. $\qquad\qquad\qquad\qquad\qquad\qquad$ □