

A Perturbative Analysis of Stochastic Descent

Samuel C. Tenka

Computer Science and AI Lab
Massachusetts Institute of Technology
Cambridge, MA 02139
`colimit@mit.edu`


June 1, 2020

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 prove that gradient noise systematically pushes SGD toward flatter minima. We characterize when and why flat minima overfit less than other minima. We generalize the Akaike Info. Criterion (AIC) to a smooth estimator of overfitting, hence enabling gradient-based model selection. We show how non-stochastic GD with a modified loss function may emulate SGD. We verify our predictions on convnets for CIFAR-10 and Fashion-MNIST.

1 Introduction

Practitioners benefit from the intuition that SGD approximates noiseless GD Bottou [1991]. In this paper, we refine that intuition by showing how gradient noise biases learning toward certain areas of weight space. 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. Leaving the rigorous development of our general theory to **APPENDIX**, our paper body highlights and intuitively discusses the theory’s main corollaries.

Our work offers a novel viewpoint of SGD as many concurrent interactions between weights and data. Diagrams such as , analogous to those of Feynman [1949], Penrose [1971], depict these interactions. In the appendix, we discuss this bridge to physics — and its relation to Hessian methods and natural GD — as topics for future research. We also discuss how this work may ameliorate or exacerbate the learning community’s disproportionate contribution to climate change. More broadly, our work adds to the body of theory on optimization in the face of uncertainty, theory that may one day inform solutions to emerging issues in user privacy and pedestrian safety.

1.1 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 . Our technical contribution is to organize the computation of this Taylor series via combinatorial objects we call *diagrams*:


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

Deferring details to later sections and appendices, we illustrate this work flow. First, let $l_x(\theta)$ be weight θ 's loss on datapoint x . We define a tensor \leftrightarrow diagram dictionary:

$$\begin{aligned} G &\triangleq \mathbb{E}_x [\nabla l_x(\theta)] \triangleq \text{red arrow} \\ H &\triangleq \mathbb{E}_x [\nabla \nabla l_x(\theta)] \triangleq \text{red double arrow} & C &\triangleq \mathbb{E}_x [(\nabla l_x(\theta) - G)^2] \triangleq \text{red arrow loop} \\ J &\triangleq \mathbb{E}_x [\nabla \nabla \nabla l_x(\theta)] \triangleq \text{red triple arrow} & S &\triangleq \mathbb{E}_x [(\nabla l_x(\theta) - G)^3] \triangleq \text{red arrow loop with tail} \end{aligned}$$

Here, G, H, J denote the loss's derivatives w.r.t. θ , and G, C, S denote the gradient's cumulants w.r.t. the randomness in x . Each $\nabla^d l_x$ corresponds to a node with d edges emanating, and fuzzy outlines group nodes that occur within the same expectation.

We may pair together the loose ends of the above (and of analogues with more edges) to obtain *diagrams*.¹ E.g., we may join $C = \text{red arrow loop}$ with $H = \text{red double arrow}$ to get . As another example, we may join two copies of $G = \text{red arrow}$ with two copies of $H = \text{red double arrow}$ to get . Intuitively, each diagram represents the interaction of its parts: of gradients (G), noise (C, S, \dots) and curvature (H, J, \dots). **APPENDIX** interprets these diagrams physically.

Example 1. Does non-Gaussian noise affect SGD? Specifically, since the skew S measures non-gaussianity, let's compute how S affects test loss. The recipe is to identify the fewest-edged diagrams containing $S = \text{red arrow loop with tail}$. In this case, there is one fewest-edged diagram — ; it results from joining S with $J = \text{red triple arrow}$. To evaluate a diagram, we multiply its components (here, S, J) with exponentiated ηH 's, one for each edge:

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

This is S 's leading order contribution to SGD's test loss written in an eigenbasis of ηH .

Remark 1. For large T and isotropic ηH , this becomes $-(\eta^3/3!) \sum_{\mu\nu\lambda} S_{\mu\nu\lambda} J_{\mu\nu\lambda} / 3\eta |H|$. Since $J = \nabla H$, $J/|H|$ measures the relative change in curvature H w.r.t. θ . So non-gaussian noise affects SGD proportion to the logarithmic derivative of curvature.

¹ A diagram's colors and geometric layout lack meaning: we **color** only for convenient reference, e.g. to a diagram's "green nodes". Only the topology of a diagram — not its size or angles — appear in our theory.

1.2 Background, notation, and assumptions

We sometimes implicitly sum repeated Greek indices: if a covector A and a vector B ¹ have coefficients A_μ, B^μ , then $A_\mu B^\mu \triangleq \sum_\mu A_\mu \cdot B^\mu$. We regard the learning rate as an inverse metric $\eta^{\mu\nu}$ that converts gradient covectors to displacement vectors [Bonnabel, 2013]. We use the learning rate η to raise indices: e.g., $H^\mu_\lambda \triangleq \eta^{\mu\nu} H_{\nu\lambda}$ and $C^\mu_\mu \triangleq \sum_{\mu\nu} \eta^{\mu\nu} \cdot C_{\nu\mu}$. Though η is a tensor, we may still define $o(\eta^d)$: a quantity q *vanishes to order* η^d when $\lim_{\eta \rightarrow 0} q/p(\eta) = 0$ for some homogeneous degree- d polynomial p .

We fix a loss function $l : \mathcal{M} \rightarrow \mathbb{R}$ on a space \mathcal{M} 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 $(l_n : 0 \leq n < N)$ 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 Appendix FILL IN’s hypotheses, e.g. that l, l_x are analytic and that moments exist. E.g., our theory models tanh networks with cross entropy loss on bounded data — with arbitrary weight sharing, skip connections, soft attention, dropout, and weight decay.

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. $E = TN/B$ epochs, i.e. $M = T/N$ updates per point) $\theta^\mu := \theta^\mu - \eta^{\mu\nu} \nabla_\nu \sum_{n \in \mathcal{B}_t} l_n(\theta)/B$, where \mathcal{B}_t is the t th batch. Our paper’s body — but not appendices — will assume **SGD has** $E = B = 1$ **and GD has** $T = B = N$ unless otherwise stated.

1.3 Related Work

Several research programs treat the overfitting of SGD-trained networks [Neyshabur et al., 2017a]. E.g., Bartlett et al. [2017] controls the Rademacher complexity of deep hypothesis classes, leading to optimizer-agnostic generalization bounds. Yet SGD-trained networks generalize despite their ability to shatter large sets [Zhang et al., 2017], so generalization must arise from not only architecture but also optimization [Neyshabur et al., 2017b]. Others approximate SGD by SDE to analyze implicit regularization (e.g. Chaudhari and Soatto [2018]), but, per Yaida [2019a], such continuous-time analyses cannot treat SGD noise correctly. We avoid these pitfalls by Taylor expanding around $\eta = 0$ as in Roberts [2018]; unlike that work, we generalize beyond order η^1 and $T = 2$.


Our theory is vacuous for large η . Other analyses treat large- η learning phenomenologically, whether by finding empirical correlates of gen. gap [Liao et al., 2018], by showing that *flat* minima generalize (Hoffer et al. [2017], Keskar et al. [2017], Wang et al. [2018]), or by showing that *sharp* minima generalize (Stein [1956], Dinh et al. [2017], Wu et al. [2018]). 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] assume uncorrelated Gaussian noise, so they cannot describe SGD’s gengap. We use Penrose diagrams to compute test losses to arbitrary order η . We allow for correlated, non-Gaussian noise and thus *any* smooth architecture. E.g., we do not assume information-geometric relationships between C and H ,² so we may model VAEs.

¹ Vectors/covectors are also called column/row vectors.

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


2 Theory, Specialized to $E = B = 1$ SGD's Test Loss

A *diagram* is a finite rooted tree equipped with a partition of its nodes obeying 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 the nodes within each part via fuzzy outlines. A diagram is *irreducible* when each of its degree-2 nodes is in a part of size one. An *embedding* f of a diagram D is an injection from the diagram'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 is increase. E.g., f might send 's

red part to $t = 3$ and its green part to $t = 4$, but not vice versa. Let $|\text{Aut}_f(D)|$ count the graph automorphisms of D that commute with f .

Up to unbiasing terms,¹ the *re-summed value* $\text{rvalue}_f(D)$ is constructed as follows.

Node rule: insert a factor a $\nabla^d l_x$ for each degree d node. **Edge rule:** for each edge whose endpoints f sends to times t, t' , insert a factor of $K^{|t'-t|-1}\eta$ where $K \triangleq (I - \eta H)$.

Outline rule: group the nodes in each part within expectation brackets \mathbb{E}_x . E.g., if f maps 's red part to time $t = T - \Delta t$, then (the red part gives S ; the green part, J):

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

In fact, we may integrate this expression per Remark 2 to recover Example 1.

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 will truncate the series to small d , thus focusing on few-edged diagrams and easing the combinatorics of embeddings.

Theorem 1 (Special Case). *For any T : for η small enough, SGD has expected test loss*

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


Theorem 2 (Long-Term Behavior near a Local Minimum). *If θ_\star locally minimizes l and for some positive form Q , $Q < \nabla^2 l_x(\theta_\star)$ for all x , then when we initialize SGD sufficiently close to θ_\star , the d th-order truncation of Theorem 1 converges as T diverges.*

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

¹ E.g., we actually define  to be the cumulant $C = \mathbb{E}_x[(\nabla l_x(\theta) - G)^2]$, not the moment $\mathbb{E}_x[(\nabla l_x(\theta))^2]$. This centering is routine (see APPENDIX), tedious to keep notating, and un-germane, so we ignore it.

2.2 Insights from the Formalism

2.2.1 SGD descends on a C -smoothed landscape and prefers minima flat w.r.t. C .



Corollary 1 (Computed from ). *Initialized at a test minimum, and run for long times $T \gg 1/\eta H$, SGD drifts with an expected time-averaged velocity of*

$$v^\lambda = \frac{\eta^3}{T} \sum_{\mu\nu} C_{\mu\nu} \frac{1}{\eta(H_{\mu\mu} + H_{\nu\nu})} J_{\mu\nu\lambda} \frac{1}{H_{\lambda\lambda}} + o(\eta^2) \quad \text{in an eigenbasis for } \eta H$$

Intuitively, $D = \text{diagram}$ contains a subdiagram $\text{diagram} = (K\eta)^2 CH$. By a routine check, $CH + o(\eta^2)$ is the loss increase upon convolving l with a C -shaped Gaussian. Since D connects the subdiagram to **to the test measurement** via 1 edge, it couples CH to l 's linear part, so it represents a displacement of θ away from high CH . In short, *SGD descends on a covariance-smoothed landscape*. That is, SGD prefers from among a valley's minima those that are flat w.r.t. C . Figure 1 (left) illustrates this intuition.

Yaïda [2019b] reports a small- T version of this result that scales with η^3 . Meanwhile, Corollary 1's large- T analysis scales with η^2 . Our analysis integrates the noise over many updates, hence amplifying the contribution of C , and experiments verify this scaling law. We do not make Wei and Schwab [2019]'s assumptions of thermal equilibrium, fast-slow mode separation, or constant covariance. This generality reveals novel dynamics: that the velocity field above is generically non-conservative (Section 3.1.2).

2.2.2 Both flat and sharp minima overfit less

Corollary 2 (from , ). *Initialize GD at a test minimum. The test-loss-increase and the gen.-gap (test minus train loss) due to training are, with errors $o(\eta^2)$ and $o(\eta^1)$:*

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

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

2.2.3 High- C regions repel small- E , B SGD

Corollary 3 (Epoch Number). *To order η^2 , $M = 1$ SGD with learning rate η has $\left(\frac{M-1}{M}\right)\left(\frac{B+1}{B}\right)\left(\frac{N}{2}\right)(\nabla_{\mu} C_{\nu}^{\nu}) G^{\mu}/2$ less test loss than $M = M$ SGD with learning rate η/M .*

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

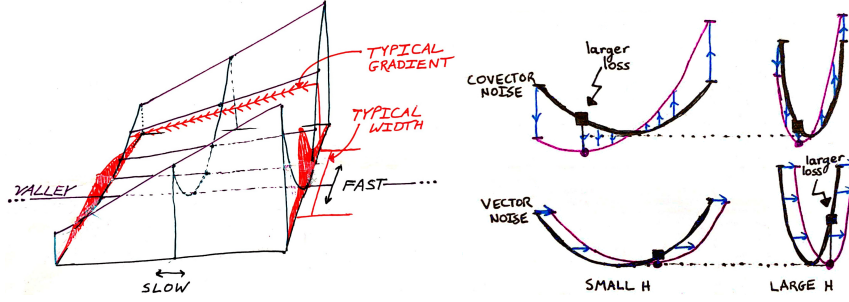




Figure 1: **Novel phenomena.** **Left:** Gradient noise induces a flow toward minima *with respect to the covariance*. Our analysis assumes neither thermal equilibrium nor fast-slow mode separation, but we label “fast and slow directions” to ease comparison with Wei and Schwab [2019]. Red densities depict a typical locations for θ in each cross-section of the valley, and the spatial variation of curvature corresponds to $J_{\mu\nu\lambda}$. **Right:** Noise structure determines how curvature affects overfitting. Geometrically, for a vector-perturbed landscape, small Hessians are favored (top row), while for covector-perturbed landscapes, large Hessians are favored (bottom row). Corollary 2 shows how the implicit regularization of fixed- ηT descent interpolates between the two rows.


2.2.4 Non-Gaussian noise affects SGD but not SDE

Stochastic Differential Equations (SDE: see Liao et al. [2018]) are a popular theoretical approximation to SGD, but SDE and SGD differ in several ways. For instance, the inter-epoch noise correlations in multi-epoch SGD measurably affect SGD’s final test loss (Corollary 3), but SDE assumes uncorrelated gradient updates. Even if we restrict to single-epoch SDE, differences arise due to time discretization and non-gaussian noise.

Corollary 5 ( , ). SGD’s test loss is $\frac{T}{2} C_{\mu\nu} H^{\mu\nu} + o(\eta^2)$ more than ODE’s and SDE’s. The deviation from SDE due to non-Gaussian noise is $-(T/6) S_{\mu\nu\lambda} J^{\mu\nu\lambda} + o(\eta^3)$.¹

3 Applying the Theory

3.1 Experiments

Our experiments’ rejection of the null hypothesis is sometimes drastic. E.g., in Figure 2  , [Chaudhari and Soatto, 2018] predicts a velocity of 0 while we predict a velocity of $\eta^2/6$. I bars and + signs to mark a 95% confidence interval based on the standard error of the mean. Appendix ?? lists architectures, procedure, and further tests.

3.1.1 Training time and batch size

We test Theorem 1’s order η^3 truncation on smooth convnets for CIFAR-10 and Fashion-MNIST. Theory agrees with experiment through on timescales long enough for accuracy

¹ This is Example 1’s more exact expression for $\eta \ll 1$: they agree to leading order in η .

to increase by 0.5% (Figure 2 $\square\square\square$, $\square\square\square$). Figure 2 $\square\square\square$ tests Corollary 4’s claim that high- C regions repel SGD more than GD. This is significant because C controls the rate at which the gengap (test minus train loss) grows (Corollary 2, Figure 2 $\square\square\square$).

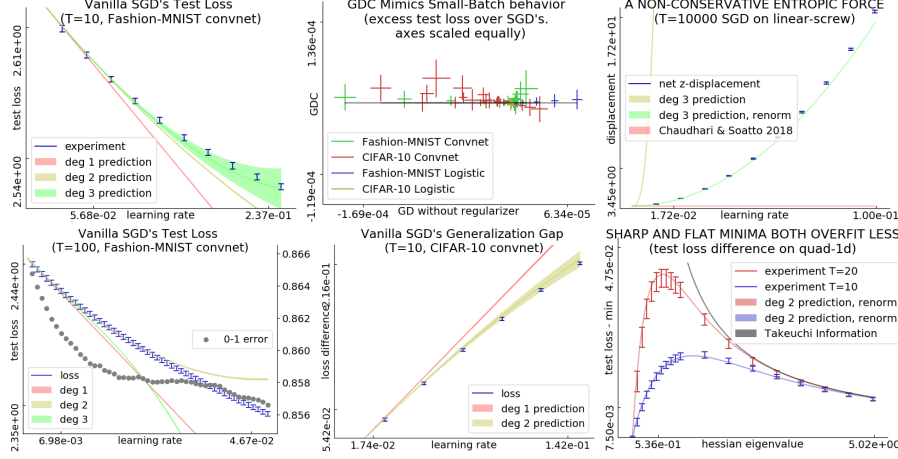



Figure 2: **Left: Perturbation models SGD for small ηT .** Fashion-MNIST convnet’s test loss vs learning rate; un-re-summed predictions. $\square\square\square$: For all init.s tested (1 shown, 11 unshown), the order 3 prediction agrees with experiment through $\eta T \approx 10^0$, corresponding to a decrease in 0-1 error of $\approx 10^{-3}$. $\square\square\square$: For large ηT , our predictions break down. Here, the order-3 prediction holds until the 0-1 error improves by $5 \cdot 10^{-3}$. **Center: C controls gen. gap and distinguishes GD from SGD.** With equal-scaled axes, $\square\square\square$ 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 init.s (zero and several Xavier-Glorot trials) for logistic regression and convnets. Here, $T = 10$. $\square\square\square$: CIFAR-10 generalization gaps. For all init.s tested (1 shown, 11 unshown), the degree-2 prediction agrees with experiment through $\eta T \approx 5 \cdot 10^{-1}$. **Right: Predictions near minima excel even for large ηT .** $\square\square\square$: On ARCHIMEDES, SGD travels the valley of global minima in the positive z direction. H and C are bounded, and the effect appears for all small η , so the effect is not a pathology of well-chosen learning rate or divergent noise. $\square\square\square$: For MEAN ESTIMATION with fixed C and a range of H s, initialized at the truth, the test losses after fixed- T optimization are smallest for very small and very large curvatures. Both sharp and flat minima overfit less.


3.1.2 Minima that are flat with respect to C attract SGD

To test Corollary 1, we construct a counter-intuitive loss landscape wherein SGD steadily moves in a direction of 0 test gradient. Our mechanism differs from that of Chaudhari and Soatto [2018]’s approximate analysis, which in this case predicts a velocity of 0.¹ Note that for fixed z , H is quadratic and S is linear. Also, since $x \sim \mathcal{N}(0, 1)$, $xS(\theta)$




¹ Indeed, our velocity is η -perpendicular to the image of $(\eta C)_v^\mu$.

has expectation 0. ARCHIMEDES thus has valley of global minima on the line $x = y = 0$. For SGD initialized at $\theta = 0$, Corollary 1 predicts a z -velocity of $+\eta^2/6$ per timestep. The prediction agrees with experiment even as the net displacement exceeds the landscape’s natural length scale of 2π (Figure 2 .

3.1.3 Sharp and flat minima both overfit less than medium minima

Prior work finds both that *sharp* minima overfit less (for, l^2 regularization sharpens minima) or that *flat* minima overfit less (for, flat minima are robust to small displacements). In fact, generalization’s relationship to curvature depends on the landscape’s noise structure (Corollary 2, Figure 2 ). To combat overfitting, we may add Corollary 2’s expression for gen. gap to l . Unlike AIC, which it subsumes, this regularizer is continuous and thus liable to descent. We call this regularizer *STIC* (APPENDIX). By descending on STIC, we may tune smooth hyperparameters such as l_2 regularization coefficients in the noisy, small- N regime $H \ll C/N$. Since matrix exponentiation takes time cubic in dimension, exact STIC is most useful for small models.

3.2 Conclusion

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 senses curvature, and curvature controls 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 propose a smooth analogue of AIC 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 [Bottou, 1991], modern infrastructure increasingly rewards large batch sizes [Goyal et al., 2018].

Since our predictions depend only on loss data near initialization, they break down after the weight moves far from initialization. Our theory thus best applies to small-movement contexts, whether for long times (large ηT) near an isolated minimum or for short times (small ηT) in general. E.g., our theory might especially illuminate meta-learners that are 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 the 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 train loss) to account for noise.

Broader Impacts

Though machine learning has the long-term potential for vast improvements in world-wide quality of life, it is today a source of enormous carbon emissions [Strubell et al., 2019]. Our analysis of SGD may lead to a reduced carbon footprint in three ways.

First, Section 2.2.3 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 small-batch SGD (e.g., Niu et al. [2011]) might require less inter-GPU communication and therefore less power.

Second, Section 3.2 discusses an application to meta-learning, which has the potential to decrease the per-task sample complexity and hence carbon footprint of modern machine learning.

Third, the generalization of AIC developed in Sections 2.2.2 and 3.1.3 permits certain forms of model selection by gradient descent rather than brute force search. This might drastically reduce the energy consumed during model selection.

That said, insofar as our theory furthers practice, it may instead contribute to the rapidly growing popularity of GPU-intensive learning, thus negating the aforementioned benefits and accelerating climate change.

More broadly, this paper analyzes optimization in the face of uncertainty. As ML systems deployed today must increasingly address *user privacy*, *pedestrian safety*, and *dataset diversity*, it becomes important to recognize that training sets and test sets differ. Toward this end, theoretical work relating to non-Gaussian noise may assist practitioners in building provably non-discriminatory, safe, or private models (e.g., Dwork et al. [2006]). By quantifying how correlated, non-Gaussian gradient noise affects descent-based learning, this paper contributes to such broader theory.

Acknowledgements

We feel deep gratitude to SHO YAJIDA, DAN A. ROBERTS, and JOSH TENENBAUM for posing some of the problems this work resolves and for their patient guidance. We appreciate the generosity of ANDY BANBURSKI, BEN R. BRAY, JEFF LAGARIAS, and WENLI ZHAO in critiquing our drafts. Without the encouragement of JASON CORSO, CHLOE KLEINFELDT, ALEX LEW, ARI MORCOS, and DAVID SCHWAB, this paper would not be. Finally, we thank our anonymous reviewers for inspiring an improved presentation. This work was funded in part by MIT’s Jacobs Presidential Fellowship and in part by Facebook AI Research.

References

- P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization algorithms on matrix manifolds, chapter 4. *Princeton University Press*, 2007.
- S.-I. Amari. Natural gradient works efficiently. *Neural Computation*, 1998.

- P.L. Bartlett, D.J. Foster, and M.J. Telgarsky. Spectrally-normalized margin bounds for neural networks. *NeurIPS*, 2017.
- S. Bonnabel. Stochastic gradient descent 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 systèmes d’équations simultanées. *Comptes rendus de l’Académie des Sciences*, 1847.
- P. Chaudhari and S. Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *ICLR*, 2018.
- Laurent Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. *ICLR*, 2017.
- C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, 2006.
- E. Dyer and G. Gur-Ari. Asymptotics of wide networks from feynman diagrams. *ICML Workshop*, 2019.
- R.P. Feynman. A space-time approach 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 obervationum erroribus minimis obnoxiae, section 39. *Proceedings of the Royal Society of Gottingen*, 1823.
- P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd. *Data @ Scale*, 2018.
- E. Hoffer, I. Hubara, and D. Soudry. Train longer, generalize better. *NeurIPS*, 2017.
- N.S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P.T.P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017.
- J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 1952.
- 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.
- 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 stochastic gradient descent. *NeurIPS*, 2011.
- A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, T. Killeen, Zeming Lin, N. Gimeshein, 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.
- 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.
- E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in nlp. *ACL*, 2019.
- 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 overparameterized 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 stochastic gradient descent. *ICLR*, 2019a.
- Sho Yaida. A first law of thermodynamics for stochastic gradient descent. *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 Appendices

These three appendices respectively serve three functions:

- to explain how to calculate using diagrams;
- to precisely state and prove our results, then pose a conjecture;
- to specify our experimental methods and results.

A How to Calculate Expected Test Losses

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

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

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

- A.1 An example calculation**
- A.2 How to identify the relevant space-time**
- A.3 How to identify the relevant diagram embeddings**
- A.4 How to evaluate each embedding**
- A.5 How to sum the embeddings' values**
- A.6 Interpreting diagrams to build intuition**
- A.7 How to solve variant problems**
- A.8 Do diagrams streamline computation?**

B Assumptions, Proofs, and Future Topics

- B.1 Setup and assumptions**
- B.2 Statements of results**
- B.3 Proof of the Dyson Lemma**
- B.4 From Dyson to diagrams**
- B.5 On Möbius inversion**
- B.6 Proof of Theorems**
- B.7 Proofs of Corollaries**
- B.8 Proofs of miscellaneous claims**
- B.9 Future Topics**

The diagram method opens the door to exploration of Lagrangian formalisms and curved backgrounds¹:

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 $\theta \leftarrow \theta - (\eta^{-1} + \lambda \nabla \nabla l_t(\theta))^{-1} \nabla l_t(\theta)$ 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.

¹ Landau and Lifshitz [1960, 1951] introduce these concepts.

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

Though our work assumes a flat metric $\eta^{\mu\nu}$, it might generalize to curved weight spaces¹. Curvature finds concrete application in the *learning on manifolds* paradigm of Absil et al. [2007], Zhang et al. [2016], notably specialized to Amari [1998]’s *natural gradient descent* and Nickel and Kiela [2017]’s *hyperbolic embeddings*. We are optimistic our formalism may resolve conjectures such as above.

C Exerimental Methods and Results

C.1 What artificial landscapes did we use?

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

GAUSS

The GAUSS landscape is a simple case of fitting a gaussian to data. In particular, it is a probability distribution \mathcal{D} over functions $l_x : \mathbb{R}^1 \rightarrow \mathbb{R}$ on 1-dimensional weight space, indexed by standard-normally distributed 1-dimensional datapoints x and defined by the expression:

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

To measure overfitting, we initialize at the true test minimum $h = 0$, then train and see how much the test loss increases.

ARCHIMEDES

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

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

Here,

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

and

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

We note that the landscape has a three-dimensional continuous screw symmetry consisting of translation along z and simultaneous rotation in $u - v$. Our experiments initialize at $x = y = z = 0$, which lies within a valley of global minima defined by $x = y = 0$.

¹ One may represent the affine connection as a node, thus giving rise to non-tensorial and hence gauge-dependent diagrams.

MEAN ESTIMATION

The MEAN ESTIMATION family of landscapes has 1 dimensional weights 1-dimensional datapoints x and defined by the expression:

$$l_x(w) \triangleq \frac{1}{2}Hw^2 + xSw$$

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 term λ as follows:

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

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

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

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 Krizhevsky [2009] and Fashion-MNIST datasets Xiao et al. [2017]. The convolutional architectures used tanh activations and Gaussian Xavier initialization. To set a standard distance scale on weight space, we parameterized the model so that the Gaussian-Xavier initialization of the linear maps in each layer differentially pulls back to standard normal initializations of the parameters.

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, e.g. 10^1 datapoints out of $\sim 10^{4.5}$ dirac masses, so overlaps are rare.

C.3 Measurement process

Diagram evaluation on real landscapes

We implemented the formulae of Appendix ?? 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.

C.4 Implementing optimizers

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

Our GDC regularizer was implemented using the unbiased estimator $\hat{C} \triangleq (l_x - l_y)_\mu l_{xy} / 2$.

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.

C.5 Software frameworks and hardware

All code and data-wrangling scripts can be found on github.com/???????/perturb. This link will be made available after the period of double-blind review.

Our code uses PyTorch 0.4.0 Paszke et al. [2019] on Python 3.6.7; there are no other substantive dependencies. The code’s randomness is parameterized by random seeds and hence reproducible.

We ran experiments on a Lenovo laptop and on our institution’s clusters; we consumed about 100 GPU-hours.

C.6 Unbiased estimators of landscape statistics

We use the following method, well known to some of our colleagues but hard to find writings on, to obtain unbiased estimates for various statistics of the loss landscape. The method is merely an elaboration of Bessel’s factor [Gauss, 1823]. For completeness, we explain it here.

Given samples from a joint probability space $\prod_{0 \leq d < D} X_d$, we seek unbiased estimates of multipoint correlators (i.e. products of expectations of products) such as $\langle x_0 x_1 x_2 \rangle \langle x_3 \rangle$. 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 $\mathbf{A}_{0 \leq s < 2S} x_0^{(s)} x_1^{(s)}$, where \mathbf{A} denotes averaging. In fact, the following also works:

$$S \left(\mathbf{A}_{0 \leq s < S} x_0^{(s)} \right) \left(\mathbf{A}_{0 \leq s < S} x_1^{(s)} \right) + (1 - S) \left(\mathbf{A}_{0 \leq s < S} x_0^{(s)} \right) \left(\mathbf{A}_{S \leq s < 2S} x_1^{(s)} \right) \quad (1)$$

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 large-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 $\mu : [D] \rightarrow [D]$ with $\mu(d) \leq d$ and $\mu \circ \mu = \mu$. Note that $|\mu| := |\text{im}(\mu)|$ counts μ ’s parts. We then define the statistic

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

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

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

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

$$\mathbb{E} \left[S^D \{x\}_\mu \right] = \sum_{\tau \leq \mu} \left(\prod_{0 \leq d < D} \frac{S!}{(S - |\tau(\mu^{-1}(d))|)!} \right) \langle x \rangle_\tau \quad (2)$$

where ‘ $\tau \leq \mu$ ’ ranges through partitions *finer* than μ , i.e. maps τ through which μ factors.

In smaller steps, 2 holds because

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

Solving 2 for $\langle x \rangle_{\mu}$, we find:

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

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

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

C.7 Additional figures

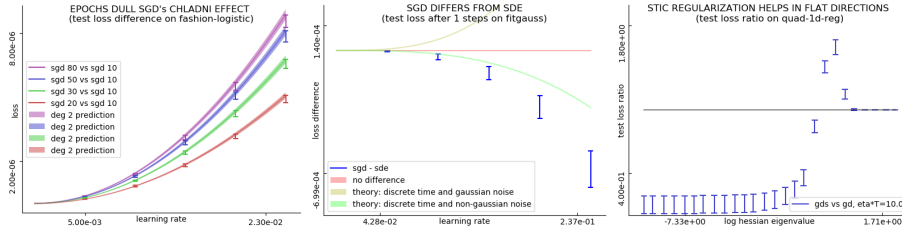


Figure 3: **Further experimental results.** **Left:** SGD with 2, 3, 5, 8 epochs incurs greater test loss than one-epoch SGD (difference shown in I bars) by the predicted amounts (predictions shaded) for a range of learning rates. Here, all SGD runs have $N = 10$; we scale the learning rate for E -epoch SGD by $1/E$ to isolate the effect of inter-epoch correlations away from the effect of larger ηT . **Center:** SGD’s difference from SDE after $\eta T \approx 10^{-1}$ with maximal coarseness on the Gaussian-fit problem. Two effects not modeled by SDE — time-discretization and non-Gaussian noise oppose on this landscape but do not completely cancel. Our theory approximates the above curve with a correct sign and order of magnitude; we expect that the fourth order corrections would improve it further. **Right:** Blue intervals regularization using Corollary 2. When the blue intervals fall below the black bar, this proposed method outperforms plain GD. For MEAN ESTIMATION with fixed C and a range of H s, 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 .

D History of SGD

We were surprised to learn of gradient descent’s long history:

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