Correlated Noise in Epoch-Based Stochastic Gradient Descent: Implications for Weight Variances

Marcel Kühn MKUEHN@ITP.UNI-LEIPZIG.DE
Institute for Theoretical Physics, University of Leipzig, 04103 Leipzig, Germany

Bernd Rosenow ROSENOW@PHYSIK.UNI-LEIPZIG.DE
Institute for Theoretical Physics, University of Leipzig, 04103 Leipzig, Germany

Abstract

Stochastic gradient descent (SGD) is a fundamental optimization method in neural networks, yet the noise it introduces is often assumed to be uncorrelated over time. This paper challenges that assumption by examining epoch-based noise correlations in discrete-time SGD with momentum under a quadratic loss. Assuming that the noise is independent of small fluctuations in the weight vector, we calculate the exact autocorrelation of the noise and find that SGD noise is anti-correlated in time. We explore the impact of these anti-correlations on SGD dynamics, finding that for directions with curvature below a hyperparameter-dependent crossover value, the weight variance is significantly reduced. This reduction leads to decreased loss fluctuations, which we relate to SGD's ability to find flat minima, thereby enhancing generalization performance.

1. Introduction

Initially developed to address the challenges of computational efficiency in neural networks, stochastic gradient descent (SGD) has exhibited exceptional effectiveness in managing large datasets compared to the full gradient methods [1]. It has since garnered widespread acclaim in the machine learning domain [2], with applications spanning image recognition [3–5], natural language processing [6, 7], and mastering complex games beyond human capabilities [8]. Alongside its numerous variants [9–11], SGD remains the cornerstone of neural network optimization.

SGD's success can be attributed to several key properties, such as rapid escape from saddle points [12] and its capacity to circumvent "bad" local minima, instead locating broad minima that generally lead to superior generalization [13–18]. This is often ascribed to anisotropic gradient noise [19–25]. Nonetheless, recent empirical research posits that even full gradient descent, with minor adjustments, can achieve generalization performance comparable to that of SGD [26].

To deepen our understanding, multiple studies have investigated the limiting dynamics of neural network weights during the latter stages of training [27, 28]. Of particular interest is the behavior of weight fluctuations in proximity to a minimum of the loss function [15, 29, 30]. Several authors found empirical evidence that the covariance matrix $\mathbf{C}$ of SGD is proportional to the Hessian matrix $\mathbf{H}$ of the loss function [18, 20–23, 31]. Consequently, theory posits that the stationary covariance matrix of weights $\mathbf{\Sigma}$ exhibits isotropy for sufficiently small learning rates [15, 28, 30]. Nevertheless, a recent empirical investigation [32] identified profound anisotropy in $\mathbf{\Sigma}$.

In this work, we delve into both theoretical and empirical analyses of weight fluctuations during the later stages of training, accounting for the emergence of anti-correlations in the noise produced by SGD, which stem from the prevalent epoch-based learning schedule. As a result of these

anti-correlations, we discover that the covariance matrix $\mathbf{\Sigma}$ displays anisotropy and is smaller than expected in a subspace of weight directions corresponding to Hessian eigenvectors with small eigenvalues (EVs), while maintaining the isotropy of $\mathbf{\Sigma}$ in directions associated with Hessian eigenvectors possessing large EVs. Our theoretical predictions are validated through the analysis of a neural network's training within a subspace of its top Hessian eigenvectors.

In addition, we demonstrate that for a small convolutional network trained in CIFAR10 the anti-correlations in SGD noise described above significantly increase the test accuracy, and by linking this result to a previous study on artificially added anti-correlated noise and its benefits [33], we argue that the anti-correlations in SGD noise suppress diffusion in flat directions, and in this way contribute to finding flatter minima with better test accuracy.

2. Background

We consider a neural network characterized by its weight vector, $\boldsymbol{\theta} \in \mathbb{R}^d$. The network is trained on a set of N training examples, each denoted by x_n, with $n = 1, ..., N$. The loss function, defined as $L(\boldsymbol{\theta}) \coloneqq \frac{1}{N}\sum_{i=1}^{N} l(\boldsymbol{\theta}, x_n)$, represents the average of individual losses incurred for each training example, $l(\boldsymbol{\theta}, x_n)$. To keep the analysis general, we consider a training process that employs stochastic gradient descent augmented with heavy ball momentum. This approach updates the network parameters according to the following rules:

$$\mathbf{g}_k(\boldsymbol{\theta}) = \frac{1}{S}\sum_{n\in\mathcal{B}_k} \boldsymbol{\nabla} l(\boldsymbol{\theta}, x_n)\ , \qquad \mathbf{v}_k = -\eta \mathbf{g}_k(\boldsymbol{\theta}_{k-1}) + \beta \mathbf{v}_{k-1}\ , \qquad \boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \mathbf{v}_k\ . \tag{1}$$

Here, k signifies the discrete update step index, η is the learning rate, and β is the momentum parameter. The stochastic gradient at each step is computed with respect to a batch of $S \ll N$ random examples. Each batch is denoted by $\mathcal{B}_k = \{n_1, ..., n_S\}$, where $n_j \in \{1, \ldots, N\}$. The training process is structured into epochs. During each epoch, every training example is used exactly once, implying that the examples are drawn without replacement within the same epoch.

In the realm of SGD as opposed to full gradient descent, we introduce noise, denoted as $\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta}) \coloneqq \mathbf{g}_k(\boldsymbol{\theta}) - \boldsymbol{\nabla} L(\boldsymbol{\theta})$, with a covariance matrix $\mathbf{C}(\boldsymbol{\theta}) \coloneqq \mathrm{cov}\big(\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta}), \boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta})\big)$. Our primary focus then lies on the asymptotic or limiting covariance matrix (see Appendix C) of the weights $\mathbf{\Sigma} \coloneqq \mathrm{cov}\big(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k\big)$ caused by the noise.

3. Anti-correlated noise and its Implications

Autocorrelation of the noise. We are considering the correlation between two noise terms which stem from different update steps. In the case of SGD, when we sample the examples without replacement while keeping the weight vector $\boldsymbol{\theta}$ constant, there are inherent anti-correlations in the noise. It follows from the definition of the noise terms $\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta}) \coloneqq \mathbf{g}_k(\boldsymbol{\theta}) - \boldsymbol{\nabla} L(\boldsymbol{\theta})$ that in this setting the sum over all noise terms of one epoch is equal to zero. This means that if at the beginning of an epoch a noise term points into one direction, we know that later noise terms from the same epoch must point into the opposite direction, hence anti-correlations emerge.

Theorem 1 *If the total number of examples N is an integer multiple of the batch size S and the parameters $\boldsymbol{\theta}$ of a network are kept fixed, then the autocorrelation formula for the gradient*

noise of an epoch-based learning schedule, where the examples for one epoch are drawn without replacement, is given by

$$\operatorname{cov}\big(\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta}), \boldsymbol{\delta}\mathbf{g}_{k+h}(\boldsymbol{\theta})\big) = \mathbf{C}(\boldsymbol{\theta}) \cdot \left(\delta_{h,0} - \mathbf{1}_{\{1,\dots,M\}}(|h|)\frac{M-|h|}{M(M-1)}\right) , \tag{2}$$

where $M := N/S$ signifies the number of batches per epoch.

In the above theorem $\mathbf{1}_A(k)$ represents the indicator function over the set A, which is one for $k \in A$ and zero otherwise and $\delta_{i,j}$ represents the Kronecker delta. The actual noise autocorrelation is illustrated in Figure 1, with the experimental details elaborated in Appendix B.2. The complete calculation is available in Appendix G. It is important to note that the above formula is only applicable for a static weight vector. Nevertheless, for later stages of training, the theoretical prediction Equation (2) still seems to be a good approximation, as evidenced by the close fit of the data in Figure 1. When we sample the examples with replacement during training, there are no anti-correlations (see Appendix I).

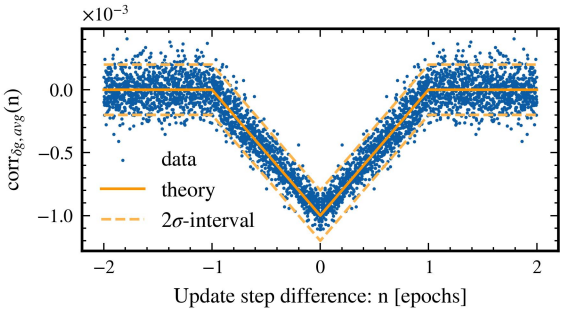

Figure 1: Autocorrelations of the SGD noise observed over a span of 20 epochs, equivalent to 20,000 update steps. This data is collected from a later phase in the training process. The autocorrelation is projected onto 5,000 Hessian eigenvectors, and the result is averaged. The theoretical prediction Equation (2) is also displayed along with a 2σ-interval, where σ represents the expected standard deviation of the SGD noise. The zero-point correlation is omitted as it is inherently equal to one.

Correlation time definition. To better understand and explain the behavior of the weight variances, we further examine the covariance matrix of the velocities $\boldsymbol{\Sigma}_{\mathbf{v}} := \operatorname{cov}\big(\mathbf{v}_k, \mathbf{v}_k\big)$, show that these two matrices commute under given assumptions, and proceed to explore their ratio $\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{\mathbf{v}}$. The EVs of this matrix ratio are denoted as τ_i (see Equation (3)). This definition aligns with that of the velocity correlation time, hence justifying the nomenclature. The equivalence stands under general assumptions (see Appendix F) that are satisfied in our problem setup of Theorem 3 (see Appendix E.4).

Theorem 2 *Under general assumptions, satisfied by the problem setup of Theorem 3, it holds that*

$$\tau_i := \frac{2\sigma_{\theta,i}^2}{\sigma_{v,i}^2} = \frac{\sum_{n=1}^{\infty} n \cdot \operatorname{cov}\big(v_{k,i}, v_{k+n,i}\big)}{\sum_{n=1}^{\infty} \operatorname{cov}\big(v_{k,i}, v_{k+n,i}\big)} , \tag{3}$$

justifying the label correlation time for this variance ratio. $\sigma_{\theta,i}^2$ and $\sigma_{v,i}^2$ are the weight and velocity variance for a projection onto a vector $\mathbf{p_i}$, with $\theta_{k,i} := \boldsymbol{\theta}_k \cdot \mathbf{p_i}$ and $v_{k,i} := \mathbf{v}_k \cdot \mathbf{p_i}$. The derivation is presented in Appendix F.

Variance for late training phase. In light of the autocorrelation of the noise calculated earlier, we want to present the expected weight and velocity variances at a later stage of the training. To describe the conditions of this phase, we adopt the following assumptions.

Assumption 1: Quadratic Approximation We postulate that we have reached a minimum point of the loss function, which can be adequately represented with a quadratic form as $L(\boldsymbol{\theta}) =$

$L_0 + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_*)^\top \mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}_*)$. We can set L_0 and $\boldsymbol{\theta}_*$ to zero without any loss of generality, which simplifies to $L(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^\top \mathbf{H}\boldsymbol{\theta}$.

Assumption 2: Anti-correlated Noise We presume that the autocorrelation of the SGD noise follows the relation previously calculated in Equation (2), even for a non static weight vector. For further motivation we refer to Appendix D.

Assumption 3: Hessian Noise Approximation We assume that the covariance of the noise commutes with the Hessian matrix, as discussed in Appendix A.1, $[\mathbf{C}, \mathbf{H}] = 0$.

Moreover, we assume that $0 \leq \beta < 1$ and $0 < \eta\lambda_i < 2(1+\beta)$ for all eigenvalues λ_i of $\mathbf{H}$. If these conditions are not met, the weight fluctuations would diverge. With the previously stated assumptions in place, the covariance matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_\mathbf{v}$ commute with $\mathbf{C}$, $\mathbf{H}$, and with each other (see Appendix E.1). As a result, they all share a common eigenbasis $\mathbf{p_i}$, with $i = 1, ..., d$, which facilitates the computation of the expected variance (see Appendix M for the case $[\mathbf{C}, \mathbf{H}] \neq 0$). We define the EVs for the common eigenvector $\mathbf{p_i}$ as λ_i for $\mathbf{H}$, $\sigma^2_{\theta,i}$ for $\boldsymbol{\Sigma}$, $\sigma^2_{v,i}$ for $\boldsymbol{\Sigma}_\mathbf{v}$, and $\sigma^2_{\delta g,i}$ for $\mathbf{C}$. We denote the number of batches per epoch as $M = N/S$, presuming it is an integer.

Theorem 3 *With the above assumptions and definitions the following relation for the weight and velocity variances hold:*

$$\begin{pmatrix} \sigma^2_{\theta,i} \\ \sigma^2_{v,i} \end{pmatrix} = \eta^2 \sigma^2_{\delta g,i} \mathbf{F_i} \left[\mathbf{e}_1 - \left(\mathbf{E_i} + \mathbf{E_i}^\top \right) \mathbf{e}_1 \right] , \tag{4}$$

where the matrices $\mathbf{F_i}$ *and* $\mathbf{E_i}$ *are explicitly expressed as:*

$$\mathbf{F_i} = \frac{1}{(1-\beta)\big(2(1+\beta) - \eta\lambda_i\big)} \begin{pmatrix} \frac{1+\beta}{\eta\lambda_i} & \frac{2\beta(\eta\lambda_i - 1 - \beta)}{\eta\lambda_i} \\ 2 & 2(\eta\lambda_i - 2) \end{pmatrix} \tag{5}$$

$$\mathbf{E_i} \coloneqq \mathbf{D_i} \frac{\mathbf{D_i}^M + (\mathbf{1} - \mathbf{D_i})M - \mathbf{1}}{(\mathbf{1} - \mathbf{D_i})^2 M(M-1)} \mathbf{e}_1 \mathbf{e}_1^\top \ , \ \mathbf{D_i} \coloneqq \begin{pmatrix} 1 + \beta - \eta\lambda_i & -\beta \\ 1 & 0 \end{pmatrix} . \tag{6}$$

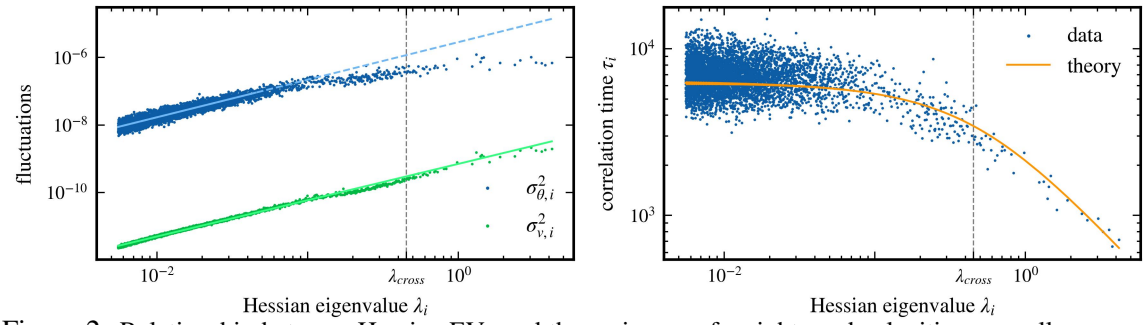

Figure 2: Relationship between Hessian EVs and the variances of weights and velocities, as well as correlation times. The mean velocity of the weight trajectory has been subtracted (see Appendix B.3). In the left panel, we present the variances of weights and velocities. The solid lines signify the regions utilized for a linear fit. The exponents resulting from the power law relationship are 1.077 ± 0.012 for weight variance and 1.066 ± 0.002 for velocity variance, with a 2σ-error. Our theory suggests these exponents should be equal to one. The right panel showcases the correlation time together with the theoretical prediction resulting from Equation (4).

The calculations can be found in Appendix E.2. The exact relation Equation (4) can be easily evaluated numerically, which shows good agreement with experimental data in Figure 2. But for a

more intuitive understanding, the relation can also be approximated by assuming that $M \gg 1/(1-\beta)$, which implies that the correlation time induced by momentum is substantially shorter than one epoch. Consequently, two distinct regimes of Hessian EVs emerge, separated by $\lambda_{\text{cross}} := 3(1-\beta)/(\eta M)$. For each of these regimes, specific simplifications apply. Notably, at λ_{cross}, both approximations converge.

Corollary 4 Relations for large Hessian EVs: *For Hessian eigenvectors with EVs $\lambda_i > \lambda_{cross}$ and when $M \gg 1/(1-\beta)$, the effects of noise anti-correlations are minimal. Consequently, we can use the following approximate relationships, which also hold true in the absence of correlations:*

$$\sigma^2_{\theta,i} \approx \frac{\eta^2 \sigma^2_{\delta g,i}}{(1-\beta)\big(2(1+\beta)-\eta\lambda_i\big)} \cdot \frac{1+\beta}{\eta\lambda_i}\,,\ \sigma^2_{v,i} \approx \frac{\eta^2 \sigma^2_{\delta g,i}}{(1-\beta)\big(2(1+\beta)-\eta\lambda_i\big)} \cdot 2\,, \tag{7}$$

and $\tau_i \approx \frac{1+\beta}{\eta\lambda_i}$. The detailed derivation of these formulas is presented in Appendix E.3.

Corollary 5 Relations for small Hessian EVs: *In the case of Hessian eigenvectors associated with EVs $\lambda_i < \lambda_{cross}$ and under the condition that $M \gg 1/(1-\beta)$, the noise anti-correlation significantly modifies the outcome. We can express the approximate relationships as follows:*

$$\sigma^2_{\theta,i} \approx \frac{\eta^2 \sigma^2_{\delta g,i}}{2(1-\beta)(1+\beta)} \cdot \frac{M}{3}\frac{1+\beta}{1-\beta}\,,\ \sigma^2_{v,i} \approx \frac{\eta^2 \sigma^2_{\delta g,i}}{2(1-\beta)(1+\beta)} \cdot 2 \tag{8}$$

and $\tau_i \approx \frac{M}{3}\frac{1+\beta}{1-\beta} =: \tau_{SGD}$. The derivation of these formulas is provided in Appendix E.3.

By considering the frequent case that the product $\eta\lambda_i$ is considerably less than one and assuming that the noise covariance matrix is proportional to the Hessian matrix, we derive the following power laws for the variances: $\sigma^2_{v,i} \propto \lambda_i$ independent of the subspace, $\sigma^2_{\theta,i} \propto$ const. for large Hessian EVs and $\sigma^2_{\theta,i} \propto \lambda_i$ for small Hessian EVs and therefore smaller than the expected isotropic variance.

4. Discussion

To further assess the impact of anti-correlations of the gradient noise we investigated the difference in generalization performance for drawing batches in SGD with and without replacement. We have trained the network described in Appendix B.1 with the same training schedule, with 20 different seeds, and have considered the maximum of the test accuracies computed after each epoch. We find that the test accuracy for training without replacement is 0.7% ± 0.2% higher than for training with replacement. The maximum test accuracy for SGD without replacement was 64.5% on average, and only 63.8% for SGD with replacement.

The above is in agreement with the results of Orvieto et al. [33], where they considered full batch gradient descent with artificially added noise that is anti-correlated in time. This noise was found to be beneficial for test accuracy and led to flatter minima. The anti-correlations considered have a very short correlation time, but are otherwise essentially equivalent to those of the SGD without replacement we have described. We therefore propose that the positive effects described by the study could be extended to SGD without replacement because of the anti-correlations.

Conclusion Our exploration of anti-correlations in SGD noise, which result from drawing examples without replacement, reveals a lower-than-expected weight variance in Hessian eigendirections with EVs smaller than the crossover value λ_{cross}. This reduced variance is beneficial because gradients in flat directions can then dominate fluctuations and lead the network towards even flatter minima with improved generalization performance.

References

[1] Léon Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[4] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. IEEE, 2016.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[8] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[9] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2017.

[11] Robin M Schmidt, Frank Schneider, and Philipp Hennig. Descending through a Crowded Valley - Benchmarking Deep Learning Optimizers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 9367–9376. PMLR, 2021.

[12] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40, pages 797–842. PMLR, 2015.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Flat Minima. *Neural Computation*, 9(1):1–42, 1997.

[14] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv preprint arXiv:1609.04836*, 2017.

[15] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three Factors Influencing Minima in SGD. *arXiv preprint arXiv:1711.04623*, 2018.

[16] Samuel L. Smith and Quoc V. Le. A Bayesian Perspective on Generalization and Stochastic Gradient Descent. *arXiv preprint arXiv:1710.06451*, 2018.

[17] Stephan Wojtowytsch. Stochastic gradient descent with noise of machine learning type. Part II: Continuous time analysis. *arXiv preprint arXiv:2106.02588*, 2021.

[18] Zeke Xie, Issei Sato, and Masashi Sugiyama. A Diffusion Theory For Deep Learning Dynamics: Stochastic Gradient Descent Exponentially Favors Flat Minima. *arXiv preprint arXiv:2002.03495*, 2021.

[19] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[20] Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical Analysis of the Hessian of Over-Parametrized Neural Networks. *arXiv preprint arXiv:1706.04454*, 2018.

[21] Yao Zhang, Andrew M. Saxe, Madhu S. Advani, and Alpha A. Lee. Energy–entropy competition and the effectiveness of stochastic gradient descent in machine learning. *Molecular Physics*, 116(21-22):3214–3223, 2018.

[22] Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George Dahl, Chris Shallue, and Roger B Grosse. Which Algorithmic Choices Matter at Which Batch Sizes? Insights From a Noisy Quadratic Model. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[23] Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The Anisotropic Noise in Stochastic Gradient Descent: Its Behavior of Escaping from Sharp Minima and Regularization Effects. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7654–7663. PMLR, 2019.

[24] Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. Hessian based analysis of SGD for Deep Nets: Dynamics and Generalization. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 190–198. Society for Industrial and Applied Mathematics, 2020.

[25] Liu Ziyin, Kangqiao Liu, Takashi Mori, and Masahito Ueda. Strength of Minibatch Noise in SGD. *arXiv preprint arXiv:2102.05375*, 2022.

[26] Jonas Geiping, Micah Goldblum, Phillip E. Pope, Michael Moeller, and Tom Goldstein. Stochastic Training is Not Necessary for Generalization. *arXiv preprint arXiv:2109.14119*, 2022.

[27] Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:1810.00004*, 2018.

[28] Daniel Kunin, Javier Sagastuy-Brena, Lauren Gillespie, Eshed Margalit, Hidenori Tanaka, Surya Ganguli, and Daniel L. K. Yamins. The Limiting Dynamics of SGD: Modified Loss, Phase-Space Oscillations, and Anomalous Diffusion. *Neural Computation*, 36(1):151–174, 2023.

[29] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic Gradient Descent as Approximate Bayesian Inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

[30] Kangqiao Liu, Liu Ziyin, and Masahito Ueda. Noise and Fluctuation of Finite Learning Rate Stochastic Gradient Descent. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 7045–7056. PMLR, 2021.

[31] Valentin Thomas, Fabian Pedregosa, Bart van Merriënboer, Pierre-Antoine Manzagol, Yoshua Bengio, and Nicolas Le Roux. On the interplay between noise and curvature and its effect on optimization and generalization. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pages 3503–3513. PMLR, 2020.

[32] Yu Feng and Yuhai Tu. The inverse variance–flatness relation in stochastic gradient descent is critical for finding flat minima. *Proceedings of the National Academy of Sciences*, 118(9), 2021.

[33] Antonio Orvieto, Hans Kersting, Frank Proske, Francis Bach, and Aurelien Lucchi. Anticorrelated noise injection for improved generalization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 17094–17116. PMLR, 2022.

[34] James Martens. New Insights and Perspectives on the Natural Gradient Method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[35] Pratik Chaudhari and Stefano Soatto. Stochastic Gradient Descent Performs Variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*. IEEE, 2018.

[36] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009.

[37] Gérard Meurant and Zdeněk Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

[38] Barak A. Pearlmutter. Fast Exact Multiplication by the Hessian. *Neural Computation*, 6(1): 147–160, 1994.

Appendix A. Related Work

A.1. Hessian Noise Approximation

The equivalence between the gradient sample covariance $\mathbf{C_0}$ and the Hessian matrix of the loss $\mathbf{H}$ is an approximation that frequently appears in the literature [15, 22, 30]. Numerous theoretical arguments have indicated that when the output of a neural network closely matches the example labels, these two matrices should be similar [15, 20, 21, 23, 34]. However, even a slight deviation between network predictions and labels can theoretically disrupt this relationship, as highlighted by Thomas et al. [31]. Despite this, empirical observations suggest a strong alignment between the gradient sample covariance and the Hessian matrix near a minimum.

Zhang et al. [22] pursued this line of thought and evaluated the assumption numerically. They examined both matrices in the context of a particular basis that presents both high and low curvature across different directions. Their findings indicated a close match between curvature and gradient variance in a given direction for a convolutional image recognition network, and a reasonably good relationship for a transformer model.

Meanwhile, Thomas et al. [31] presented both a theoretical argument and empirical evidence across different architectures for image recognition. Although they did not discover an exact match between the two matrices, they did observe a proportionality between them, indicated by a high cosine similarity.

Xie et al. [18] also conducted an investigation with an image recognition network. In the eigenspace of the Hessian matrix, they plotted entries within a specific interval against corresponding entries from the gradient sample covariance and found a close match.

For our theoretical considerations, it is crucial to assume that both matrices commute, $[\mathbf{C_0}, \mathbf{H}] = 0$. Additionally, we hypothesize that $\mathbf{C_0} \propto \mathbf{H}$, to derive power law predictions for the weight variance.

A.2. Limiting Dynamics and Weight Fluctuations

Various studies have scrutinized the limiting dynamics, often modeling SGD as a stochastic differential equation (SDE). Commonly, researchers such as Mandt et al. [29] and Jastrzębski et al. [15] approximate the loss near a minimum as a quadratic function and present the SDE as a multivariate Ornstein-Uhlenbeck (OU) process. This process proposes a stationary weight distribution with Gaussian weight fluctuations. Jastrzębski et al. [15] further assume the Hessian noise approximation, observing under these conditions that the weight fluctuations are isotropic. Kunin et al. [28] who also incorporate momentum into their analysis, predict and empirically verify isotropic weight fluctuations. Chaudhari & Soatto [35] also investigate the SDE but without the assumption of a quadratic loss, nor that it reached equilibrium. They gain insights via the Fokker-Plank equation.

Alternatively, some studies derive relations from a stationarity assumption instead of a continuous time approximation [25, 27, 30]. Yaida [27] assumes that the weight trajectory follows a stationary distribution and derives general fluctuation-dissipation relations from that. Liu et al. [30] go further to assume a quadratic loss function, leading them to derive exact relations for the weight variance of SGD with momentum. If the additional Hessian noise approximation is made, their results also predict the weight variance to be approximately isotropic, except in directions where the product of learning rate and Hessian eigenvalue is significantly high.

Such computed weight variances are explicitly applied in various contexts, such as computing the escape rate from a minimum or assessing the approximation error in SGD, which captures the additional training error attributed to noise [30].

Feng & Tu [32] present a phenomenological theory based on their empirical findings, which also account for flat directions, unlike Kunin et al. [28]. They describe a general inverse variance-flatness relation, analyzing the weight trajectory of different image recognition networks via principal component analysis. They discovered a power law relationship between the curvature of the loss and the weight variance $\sigma^2_{\theta,i}$ in any given direction, where a higher curvature corresponds to a higher variance. They also observed that both the velocity variance $\sigma^2_{v,i}$ and the correlation time τ_i are larger for higher curvatures.[1]

In our approach, we do not make the continuous time approximation but base our results on the assumption that the weights adhere to a stationary distribution near a quadratic minimum.

Appendix B. Numerics

B.1. Analysis Setup

In order to corroborate our theoretical findings, we have conducted a small-scale experiment. We have trained a LeNet architecture, similar to the one described in [32], using the CIFAR10 dataset [36]. LeNet is a compact convolutional network comprised of two convolutional layers followed by three dense layers. The network comprises approximately 137,000 parameters. As our loss function, we employed Cross Entropy, along with an L2 regularization with a prefactor of 10^{-4}. In the main text we present results for a single seed and specific hyperparameters. However, we have also performed tests with different seeds and combinations of hyperparameters, all of which showed comparable qualitative behavior (see Appendix K). Furthermore, in Appendix L we studied a ResNet architecture [5], a different and more modern network, where we obtained similar results.

We used SGD to train the network for 100 epochs, employing an exponential learning rate schedule that reduces the learning rate by a factor of 0.98 each epoch. The initial learning rate is set at $5 \cdot 10^{-3}$, which eventually reduces to approximately $7 \cdot 10^{-4}$ after 100 epochs. The momentum parameter and the minibatch size S are set to 0.9 and 50, respectively, which results in a thousand minibatches per epoch, $M = 1000$. This setup achieves 100% training accuracy and 63% testing accuracy. The evolution of loss and accuracy during training can be seen in Appendix J. We then compute the variances right after the initial schedule over a period of 20 additional epochs, equivalent to 20,000 update steps. Throughout this analysis period, the learning rate is maintained at $7 \cdot 10^{-4}$ and the recorded weights are designated by $\boldsymbol{\theta}_k$, with $k = K, \ldots, K + T$ and $T = 20{,}000$.

Given the impracticability of obtaining the full covariance matrix for all weights and biases over this period due to the excessive memory requirements, we limit our analysis to a specific subspace. We approximate the five thousand largest eigenvalues and their associated eigenvectors of the Hessian matrix $\mathbf{H}(\boldsymbol{\theta}_K)$, drawn from the roughly 137,000 total, using the Lanczos algorithm [37]. Here, $\boldsymbol{\theta}_K$ represents the weights at the beginning of the analysis period. To compute the Hessian-vector products required for the Lanczos algorithm, we employ the resource-efficient Pearlmutter trick [38]. The eigenvectors of the Hessian matrix are represented by $\mathbf{p_i}$, and the projected weights

1. Our findings for the three quantities contrast with this previous empirical study [32] to a certain extend. In Appendix H, we clarify how the different analysis method used in that study impacts the results due to finite size effects.

by $\theta_{k,i} = \boldsymbol{\theta}_k \cdot \mathbf{p_i}$. The variances are computed exclusively for these particular directions. The distribution of the approximated 5,000 eigenvalues is illustrated in Figure 3.

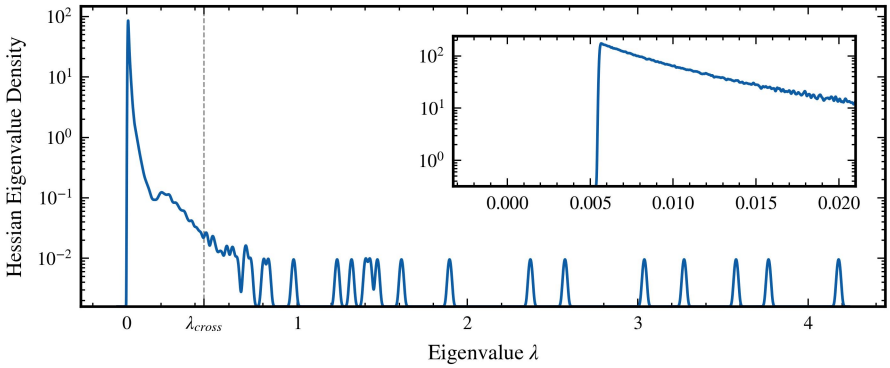

Figure 3: The distribution of the approximated 5,000 Hessian eigenvalues of the LeNet discussed in the main text. The inset shows that the smallest approximated eigenvalue has a magnitude of about 0.005.

B.2. Noise Autocorrelations

We scrutinize the correlations of noise by recording both the minibatch gradient $\mathbf{g}_k(\boldsymbol{\theta}_k)$ and the total gradient $\boldsymbol{\nabla} L(\boldsymbol{\theta}_k)$ at each update step throughout the analysis period, enabling us to capture the actual noise term $\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta}_k)$. All these are projected onto the approximated Hessian eigenvectors.

The theoretical prediction for the anti-correlation of the noise is proportional to the inverse of the number of batches per epoch, in our case on the order of 10^{-3}. To extract the predicted relationship from the fluctuating data, we compute the autocorrelation of the noise term for each individual Hessian eigenvector. We then proceed to average these results across the 5,000 approximated eigenvectors. Figure 1 provides a visual representation of this analysis, showcasing a strong alignment between the empirical autocorrelation of noise and the prediction derived from our theory. This consistency can be interpreted as a validation of our assumption that the noise is spatially independent.

B.3. Variances and Correlation time

Previous studies have observed that network weights continue to traverse the parameter space even after the loss appears to have stabilized [19, 28, 32]. This behavior persists despite the use of L2 regularization and implies that the recorded weights, $\boldsymbol{\theta}_k$, do not settle into a stationary distribution. Notably, however, over the course of the 20 epochs under scrutiny, the weight movement, excluding the SGD noise, appears to be approximately linear in time. This suggests that the mean velocity $\bar{\mathbf{v}} := \langle \mathbf{v}_k \rangle$ is substantial compared to the SGD noise. To isolate this ongoing movement and uncover the underlying structure, we redefine $\boldsymbol{\theta}_k$ and $\mathbf{v}_k$ by subtracting the mean velocity. This results in $\boldsymbol{\theta}_k^{(\mathrm{s})} := \boldsymbol{\theta}_k - \bar{\mathbf{v}} \cdot k$ and $\mathbf{v}_k^{(\mathrm{s})} := \mathbf{v}_k - \bar{\mathbf{v}}$. We then compute the variances of these redefined values, $\boldsymbol{\theta}_k^{(\mathrm{s})}$ and $\mathbf{v}_k^{(\mathrm{s})}$, which exhibit a more stationary distribution.

Again, we limit our variance calculations to the directions of the 5,000 approximated Hessian eigenvectors. In the two different regimes of Hessian eigenvalues, either greater or lesser than

the crossover value λ_{cross}, the weight and velocity variance closely follow the respective power law predictions from our theory (see upper panel of Figure 2). The slight discrepancy, where the predicted exponent of one does not lie within the error bars, may arise from minor deviations in the noise covariance from the Hessian approximation $\mathbf{C} \propto \mathbf{H}$. The calculated correlation time, derived from the ratio between the weight and velocity variance, aligns reasonably well with our theoretical predictions (see lower panel of Figure 2). This correlation time prediction remains independent of the exact relation between $\mathbf{C}$ and $\mathbf{H}$, thereby providing a more general result.

Appendix C. Definition of Limiting Quantities

When we speak of a covariance matrix or an average in the main text and in the following sections of the appendix, we mean the limiting average or the limiting covariance, unless otherwise specified. In other words, we are interested in the average of a quantity over one infinite run of SGD optimization, not the mean value for a fixed update step k averaged over multiple runs of SGD optimization. With this in mind, we define the covariance matrix of two quantities $\mathbf{a}_k$ and $\mathbf{b}_k$ as

$$\operatorname{cov}(\mathbf{a}_k, \mathbf{b}_k) \coloneqq \left\langle (\mathbf{a}_k - \langle \mathbf{a}_k \rangle_k) (\mathbf{b}_k - \langle \mathbf{b}_k \rangle_k)^\top \right\rangle_k \tag{9}$$

and the limiting average is defined as

$$\langle \mathbf{a}_k \rangle_k = \lim_{K\to\infty} \frac{1}{K+1} \sum_{k=k_0}^{k_0+K} \mathbf{a}_k \,. \tag{10}$$

When possible, we will suppress k and denote the average as $\langle \cdot \rangle$. The average is independent of the starting value k_0, therefore we can shift indices within the average, meaning $\langle \mathbf{a}_k \rangle = \langle \mathbf{a}_{k+l} \rangle$ for any $l \in \mathbb{Z}$.

To see this we take any integer $l \in \mathbb{Z}$ and instead of adding it to the index k we can also subtract it from the starting value k_0 and then separate the sum into two sums,

$$\begin{aligned}
\langle \mathbf{a}_{k+l} \rangle &= \lim_{K\to\infty} \frac{1}{K+1} \sum_{k=k_0}^{k_0+K} \mathbf{a}_{k+l} \\
&= \lim_{K\to\infty} \frac{1}{K+1} \sum_{k=k_0-l}^{k_0-l+K} \mathbf{a}_k \\
&= \lim_{K\to\infty} \frac{1}{K+1} \sum_{k=k_0-l}^{k_0-1} \mathbf{a}_k + \lim_{K\to\infty} \frac{1}{K+1} \sum_{k=k_0}^{k_0-l+K} \mathbf{a}_k \,.
\end{aligned} \tag{11}$$

The first sum is independent of K except for the factor $\frac{1}{K+1}$, so the limit of the first part is zero. The second part of the limit can be rearranged as follows,

$$\begin{aligned}
\lim_{K\to\infty} \frac{1}{K+1} \sum_{k=k_0}^{k_0-l+K} \mathbf{a}_k &= \lim_{K\to\infty} \frac{K-l+1}{K+1} \frac{1}{K-l+1} \sum_{k=k_0}^{k_0-l+K} \mathbf{a}_k \\
&= \lim_{\tilde{K}\to\infty} \frac{1}{\tilde{K}+1} \sum_{k=k_0}^{k_0+\tilde{K}} \mathbf{a}_k \qquad (12) \\
&=: \langle \mathbf{a}_k \rangle \,, \qquad (13)
\end{aligned}$$

where we used $\lim_{K\to\infty} \frac{K-l+1}{K+1} = 1$ and renamed $K - l$ in the second to last step. All together this gives us the desired relation $\langle \mathbf{a}_k \rangle = \langle \mathbf{a}_{k+l} \rangle$.

If one were to consider the covariance for a fixed update step k averaged over multiple runs of SGD optimization, it is possible that this covariance could depend on the index k, but this is not our case of interest.

Appendix D. Weight independent Noise

It is important to note that Equation (2), the formula for the autocorrelation of the noise described in Theorem 1 and derived in Appendix G, is strictly speaking only valid for a static weight vector. During training, the weights change with each update step, which could potentially change this relationship. However, if the noise terms were independent of the weight vector, the formula would still hold. A simple case where this assumption of weight independent noise holds is a model where not only the total loss can be described by a quadratic function,

$$L(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^\top \mathbf{H} \boldsymbol{\theta} + \text{const.}\,, \tag{14}$$

but the loss for a single example is described by a quadratic function as well, with the same Hessian,

$$l(\boldsymbol{\theta}, \mathbf{x}_n) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu}_n)^\top \mathbf{H} (\boldsymbol{\theta} - \boldsymbol{\mu}_n) + \text{const.}\,. \tag{15}$$

However, in this model, the minimum of the loss function for this single example is offset by an example-dependent vector. This offset results in a noise term that is independent of the current weight vector for each individual example,

$$\begin{aligned} \boldsymbol{\nabla}\big(l(\boldsymbol{\theta}, \mathbf{x}_n) - L(\boldsymbol{\theta})\big) &= -\mathbf{H}\boldsymbol{\mu}_n \\ &\neq f(\boldsymbol{\theta})\,. \end{aligned} \tag{16}$$

Therefore, also the noise introduced by SGD in this model would be independent of the current weight vector. For the later stages of training, such an assumption for the noise could be a good approximation, as evidenced by the close fit of the data to the theory in Figure 1.

Appendix E. Variance Calculation

E.1. Commutativity of the covariance matrices

In this section, we show that if $[\mathbf{C}, \mathbf{H}] = 0$ also $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_\mathbf{v}$ will commute with $\mathbf{C}$, with $\mathbf{H}$ and with each other. We make the assumptions one to three of Theorem 3 and therefore the SGD update equations become

$$\mathbf{v}_k = -\eta \mathbf{H} \boldsymbol{\theta}_{k-1} + \beta \mathbf{v}_{k-1} - \eta \boldsymbol{\delta}\mathbf{g}_k\,, \tag{17}$$

$$\boldsymbol{\theta}_k = (\mathbf{1} - \eta \mathbf{H})\,\boldsymbol{\theta}_{k-1} + \beta \mathbf{v}_{k-1} - \eta \boldsymbol{\delta}\mathbf{g}_k\,, \tag{18}$$

which can be rewritten by using the vector $\mathbf{y}_k \coloneqq \begin{pmatrix} \boldsymbol{\theta}_k & \mathbf{v}_k \end{pmatrix}^\top$, combining both the current weight and velocity variable, to be

$$\mathbf{y}_{k+1} = \mathbf{X}\mathbf{y}_k - \mathbf{z}_{k+1}\,. \tag{19}$$

Here, $\mathbf{z}_k \coloneqq \begin{pmatrix}\eta\boldsymbol{\delta}\mathbf{g}_k & \eta\boldsymbol{\delta}\mathbf{g}_k\end{pmatrix}^\top$ contains the current noise term, and the matrix governing the deterministic part of the update is defined to be

$$\mathbf{X} \coloneqq \begin{pmatrix}\mathbf{1}-\eta\mathbf{H} & \beta\mathbf{1} \\ -\eta\mathbf{H} & \beta\mathbf{1}\end{pmatrix} . \tag{20}$$

By iteratively applying Equation (19) we obtain

$$\mathbf{y}_{k+h} = \mathbf{X}^h\mathbf{y}_k - \sum_{i=1}^{h}\mathbf{X}^{h-i}\mathbf{z}_{k+i} . \tag{21}$$

Under the assumption $0 \leq \beta < 1$ and $0 < \eta\lambda_i < 2(1+\beta)$, for all eigenvalues λ_i of $\mathbf{H}$, the magnitude of the eigenvalues of $\mathbf{X}$ will be less than one. It is straightforward to show this relation for the eigenvalues of $\mathbf{X}$ by using the eigenbasis of $\mathbf{H}$. Therefore,

$$\lim_{h\to\infty}\mathbf{X}^h\mathbf{y}_k = 0. \tag{22}$$

As we can shift the index in the weight variance, h can be chosen arbitrarily large, which yields the following relation for the covariance

$$\langle\mathbf{y}_k\mathbf{y}_k^\top\rangle = \lim_{h\to\infty}\sum_{i,j=1}^{h}\mathbf{X}^{h-i}\langle\mathbf{z}_{k+i}\mathbf{z}_{k+j}^\top\rangle\left(\mathbf{X}^{h-j}\right)^\top . \tag{23}$$

Because Equation (21) together with Equation (22) implies $\langle\mathbf{y}_k\rangle = 0$ and therefore $\langle\boldsymbol{\theta}_k\rangle = 0$ and $\langle\mathbf{v}_k\rangle = 0$, the left hand side of Equation (23) contains the covariance matrices of interest,

$$\langle\mathbf{y}_k\mathbf{y}_k^\top\rangle = \begin{pmatrix}\boldsymbol{\Sigma} & \langle\boldsymbol{\theta}_k\mathbf{v}_k^\top\rangle \\ \langle\mathbf{v}_k\boldsymbol{\theta}_k^\top\rangle & \boldsymbol{\Sigma}_\mathbf{v}\end{pmatrix} . \tag{24}$$

From Equation (23) we can also infer that $\langle\mathbf{y}_k\mathbf{y}_k^\top\rangle$ is finite as the magnitude of the eigenvalues of $\mathbf{X}$ is less than one. Consequently, by Equation (24), the covariance matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_\mathbf{v}$ are finite as well. The average over the noise terms $\mathbf{z}_k$ on the right hand side of Equation (23) is by assumption equal to

$$\langle\mathbf{z}_{k+i}\mathbf{z}_{k+j}^\top\rangle = \eta^2\left(\delta_{i,j} - \mathbf{1}_{\{1,\dots,M\}}(|i-j|)\frac{M-|i-j|}{M(M-1)}\right)\cdot\begin{pmatrix}\mathbf{C} & \mathbf{C} \\ \mathbf{C} & \mathbf{C}\end{pmatrix} , \tag{25}$$

from which it follows that for any finite h every matrix entry of the two by two super matrix on the right hand side of Equation (23) is a function of $\mathbf{C}$ and $\mathbf{H}$. Therefore, when considering the limit $h \to \infty$, $[\mathbf{C}, \mathbf{H}] = 0$ implies that $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_\mathbf{v}$ will also commute with $\mathbf{C}$, with $\mathbf{H}$ and with each other.

E.2. Proof of the variance formula for one specific eigenvalue

Since $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_\mathbf{v}$ will commute with $\mathbf{C}$, with $\mathbf{H}$ and with each other, it is sufficient to prove the one dimensional case. For the multidimensional case simply apply the proof in the direction of each common eigenvector individually. The expectation values discussed below are computed with respect to the asymptotic distributions of θ and v, since we are only interested in the asymptotic

behavior of training. We want to find $\sigma_\theta^2 \coloneqq \langle \theta_k \theta_k \rangle$ and $\sigma_v^2 \coloneqq \langle v_k v_k \rangle$. We assume $0 \leq \beta < 1$ and $0 < \eta\lambda < 2(1+\beta)$ where λ is the hessian eigenvalue.

The equations describing SGD in one dimension are:

$$g_k(\theta) = \frac{\partial}{\partial \theta} L(\theta) + \delta g_k(\theta) \tag{26}$$

$$v_k = -\eta g_k(\theta_{k-1}) + \beta v_{k-1} \tag{27}$$

$$\theta_k = \theta_{k-1} + v_k \ . \tag{28}$$

Our remaining assumptions can then be described the following way

$$L(\theta) = \frac{1}{2}\theta\lambda\theta \tag{29}$$

$$\delta g_k(\theta) = \delta g_k \ \ldots \text{ is independent of } \theta \tag{30}$$

$$\Rightarrow \langle \delta g_k \delta g_{k+h} \rangle = \sigma_{\delta g}^2 \left(\delta_{h,0} - \mathbf{1}_{\{1,\ldots,M\}}(|h|) \frac{M - |h|}{M(M-1)} \right) \tag{31}$$

$$\sigma_{\delta g}^2 \coloneqq \langle \delta g_k \delta g_k \rangle \ . \tag{32}$$

With these assumptions the update equations can be described by a discrete stochastic linear equation of second order

$$\theta_k = (1 + \beta - \eta\lambda)\theta_{k-1} - \beta\theta_{k-2} - \eta\delta g_k \tag{33}$$

which can be rewritten into matrix form as follows

$$\mathbf{x}_k = \mathbf{D}\mathbf{x}_{k-1} - \eta\delta g_k \mathbf{e}_1 \tag{34}$$

$$\mathbf{x}_k \coloneqq \begin{pmatrix} \theta_k \\ \theta_{k-1} \end{pmatrix} \tag{35}$$

$$\mathbf{e}_1 \coloneqq \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{36}$$

$$\mathbf{D} \coloneqq \begin{pmatrix} 1 + \beta - \eta\lambda & -\beta \\ 1 & 0 \end{pmatrix} \ . \tag{37}$$

We are now interested in the following covariance matrix

$$\begin{aligned} \tilde{\mathbf{\Sigma}} &\coloneqq \left\langle \mathbf{x}_k \mathbf{x}_k^\top \right\rangle \\ &= \begin{pmatrix} \sigma_\theta^2 & \langle \theta_k \theta_{k-1} \rangle \\ \langle \theta_k \theta_{k-1} \rangle & \sigma_\theta^2 \end{pmatrix} \end{aligned} \tag{38}$$

where the second equality is due to the fact that $\langle \theta_k \theta_k \rangle = \langle \theta_{k-1} \theta_{k-1} \rangle$. As we are interested in the asymptotic covariance, this expectation value is independent of any finite shift of the index k. By inserting Equation (34) into $\left\langle \mathbf{x}_k \mathbf{x}_k^\top \right\rangle$ we arrive at the following equality

$$\left\langle \mathbf{x}_k \mathbf{x}_k^\top \right\rangle = \mathbf{D} \left\langle \mathbf{x}_{k-1} \mathbf{x}_{k-1}^\top \right\rangle \mathbf{D}^\top + \eta^2 \left\langle \delta g_k \delta g_k \right\rangle \mathbf{e}_1 \mathbf{e}_1^\top - \eta \left(\mathbf{D} \left\langle \mathbf{x}_{k-1} \delta g_k \right\rangle \mathbf{e}_1^\top + \left(\mathbf{D} \left\langle \mathbf{x}_{k-1} \delta g_k \right\rangle \mathbf{e}_1^\top \right)^\top \right) \tag{39}$$

which can be simplified to the equivalent equation

$$\tilde{\mathbf{\Sigma}} - \mathbf{D}\tilde{\mathbf{\Sigma}}\mathbf{D}^\top = \eta^2\sigma_{\delta g}^2 \mathbf{e}_1\mathbf{e}_1^\top - \eta\left(\mathbf{D}\left\langle \mathbf{x}_{k-1}\delta g_k\right\rangle \mathbf{e}_1^\top + \left(\mathbf{D}\left\langle \mathbf{x}_{k-1}\delta g_k\right\rangle \mathbf{e}_1^\top\right)^\top\right) . \tag{40}$$

If we apply the left-hand side on the vector $\mathbf{e}_1$, it can be expressed as

$$\left[\tilde{\mathbf{\Sigma}} - \mathbf{D}\tilde{\mathbf{\Sigma}}\mathbf{D}^\top\right]\mathbf{e}_1 = \mathbf{F}_1^{-1}\tilde{\mathbf{\Sigma}}\mathbf{e}_1 \tag{41}$$

$$\mathbf{F}_1^{-1} \coloneqq \begin{pmatrix} \eta\lambda(2-\eta\lambda) - 2\beta(1+\beta-\eta\lambda) & 2\beta(1+\beta-\eta\lambda) \\ -(1+\beta-\eta\lambda) & 1+\beta \end{pmatrix} . \tag{42}$$

Also notice $v_k = \theta_k - \theta_{k-1}$ and therefore

$$\sigma_v^2 = 2\sigma_\theta^2 - 2\langle\theta_k\theta_{k-1}\rangle , \tag{43}$$

again due to the fact that the expectation value does not depend on k. Hence, the variances can then be expressed as

$$\begin{pmatrix}\sigma_\theta^2 \\ \sigma_v^2\end{pmatrix} = \mathbf{F}_2\tilde{\mathbf{\Sigma}}\mathbf{e}_1 \tag{44}$$

$$\mathbf{F}_2 \coloneqq \begin{pmatrix} 1 & 0 \\ 2 & -2 \end{pmatrix} . \tag{45}$$

We define the matrix $\mathbf{F} \coloneqq \mathbf{F}_2\mathbf{F}_1$. By applying both sides of Equation (40) to the vector $\mathbf{e_1}$, then multiplying by the matrix $\mathbf{F}$ from the left and using Equations (41) and (44) we obtain

$$\begin{pmatrix}\sigma_\theta^2 \\ \sigma_v^2\end{pmatrix} = \mathbf{F}\left[\eta^2\sigma_{\delta g}^2 \mathbf{e}_1\mathbf{e}_1^\top - \eta\left(\mathbf{D}\left\langle \mathbf{x}_{k-1}\delta g_k\right\rangle \mathbf{e}_1^\top + \left(\mathbf{D}\left\langle \mathbf{x}_{k-1}\delta g_k\right\rangle \mathbf{e}_1^\top\right)^\top\right)\right]\mathbf{e}_1 . \tag{46}$$

with

$$\mathbf{F} = \frac{1}{(1-\beta)\big(2(1+\beta)-\eta\lambda)\big)}\begin{pmatrix} \frac{1+\beta}{\eta\lambda} & \frac{2\beta(\eta\lambda-1-\beta)}{\eta\lambda} \\ 2 & 2(\eta\lambda-2) \end{pmatrix} . \tag{47}$$

To simplify Equation (46) further we go back to Equation (34) and iterate it to obtain

$$\mathbf{x}_k = \mathbf{D}^n\mathbf{x}_{k-n} - \eta\sum_{h=0}^{n-1}\mathbf{D}^h\mathbf{e}_1\delta g_{k-h} . \tag{48}$$

We note that $\langle\mathbf{x}_{k-n}\delta g_k\rangle = 0$ for $n \geq M$. The correlation between noise terms separated by at least one epoch vanishes, and $\mathbf{x}_k$ only depends on past noise terms. By setting $n = M$ we find

$$\begin{aligned}\left\langle \mathbf{x}_{k-1}\delta g_k\right\rangle &= \mathbf{D}^M\left\langle \mathbf{x}_{k-1-M}\delta g_k\right\rangle - \eta\sum_{h=0}^{M-1}\mathbf{D}^h\mathbf{e}_1\left\langle \delta g_k\delta g_{k-1-h}\right\rangle \\ &= -\eta\sigma_{\delta g}^2\sum_{h=0}^{M-1}\mathbf{D}^h\mathbf{e}_1\left(-\frac{M-(h+1)}{M(M-1)}\right) ,\end{aligned} \tag{49}$$

where the assumption about the correlation of the noise terms, Equation (31), was inserted for the last line. Equation (49) is a sum of a finite geometric series and a derivative of that which can be simplified to

$$\langle \mathbf{x}_{k-1} \delta g_k \rangle = \eta \sigma_{\delta g}^2 \frac{\mathbf{D}^M + (\mathbf{1} - \mathbf{D})M - \mathbf{1}}{(\mathbf{1} - \mathbf{D})^2 M(M-1)} \mathbf{e}_1 \,. \tag{50}$$

Substituting this result back into Equation (46) yields

$$\begin{pmatrix} \sigma_\theta^2 \\ \sigma_v^2 \end{pmatrix} = \eta^2 \sigma_{\delta g}^2 \mathbf{F} \left[\mathbf{e}_1 - \left(\mathbf{E} + \mathbf{E}^\top \right) \mathbf{e}_1 \right] \tag{51}$$

with the definition

$$\mathbf{E} \coloneqq \mathbf{D} \frac{\mathbf{D}^M + (\mathbf{1} - \mathbf{D})M - \mathbf{1}}{(\mathbf{1} - \mathbf{D})^2 M(M-1)} \mathbf{e}_1 \mathbf{e}_1^\top \,. \tag{52}$$

With Equation (51) we have arrived at the exact formula for the variances which can easily be evaluated numerically.

E.3. Approximation of the exact formula

It is possible to approximate the exact result for the variance assuming small or large eigenvalues, respectively. For that, it is necessary to approximate $\mathbf{D}^M \mathbf{e_1}$. To do so, we will use the the following eigendecomposition of $\mathbf{D}$

$$\mathbf{D} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^{-1} \tag{53}$$

$$\mathbf{\Lambda} = \begin{pmatrix} \Lambda_+ & 0 \\ 0 & \Lambda_- \end{pmatrix} \tag{54}$$

$$\mathbf{Q} = \begin{pmatrix} \Lambda_+ & \Lambda_- \\ 1 & 1 \end{pmatrix} \tag{55}$$

$$\mathbf{Q}^{-1} = \frac{1}{\Lambda_+ - \Lambda_-} \begin{pmatrix} 1 & -\Lambda_- \\ -1 & \Lambda_+ \end{pmatrix} \tag{56}$$

$$\Lambda_\pm = \frac{1}{2} \left(1 + \beta - \eta\lambda \pm s \right) \tag{57}$$

$$s \coloneqq \sqrt{(1-\beta)^2 - \eta\lambda\big(2(1+\beta) - \eta\lambda\big)} \tag{58}$$

It is straightforward to show that the magnitude of the eigenvalues of $\mathbf{D}$ is strictly smaller than one, $|\Lambda_\pm| < 1$, under the conditions $0 < \eta\lambda < 2(1+\beta)$ and $0 \leq \beta < 1$.

LARGE HESSIAN EIGENVALUES

$$\sigma_\theta^2 \approx \frac{\eta^2 \sigma_{\delta g}^2}{(1-\beta)(2(1+\beta) - \eta\lambda)} \cdot \frac{1+\beta}{\eta\lambda} \tag{59}$$

$$\sigma_v^2 \approx \frac{\eta^2 \sigma_{\delta g}^2}{(1-\beta)(2(1+\beta) - \eta\lambda)} \cdot 2 \tag{60}$$

We will show that this approximation for large hessian eigenvalues is valid under the assumption $M(\eta\lambda)^2 \gg 1$ where M is the number of batches per epoch. However, numerical studies indicate that these relations also hold under the previously mentioned assumptions of $\frac{M\eta\lambda}{1-\beta} \gg 1$, equivalent to $\lambda \gtrsim \lambda_{\text{cross}}$, and $M(1-\beta) \gg 1$.

Inserting the eigendecomposition of $\mathbf{D}$ into the expression $\mathbf{D}^M\mathbf{e_1}$ yields

$$\mathbf{D}^M\mathbf{e_1} = \begin{pmatrix} y_{M+1} \\ y_M \end{pmatrix} \tag{61}$$

$$y_M \coloneqq \frac{\Lambda_+^M - \Lambda_-^M}{\Lambda_+ - \Lambda_-}. \tag{62}$$

From the definition of y_M one sees that

$$y_M = \frac{\Lambda_+ + \Lambda_-}{2}\, y_{M-1} + \frac{\Lambda_+^{M-1} + \Lambda_-^{M-1}}{2}\,, \tag{63}$$

and by using $|\Lambda_\pm| < 1$ as well as $y_0 = 0$ one can show iteratively that

$$|y_M| \leq M + 1. \tag{64}$$

Therefore, we have

$$\left\|\mathbf{D}^M\mathbf{e_1}\right\|_\infty \leq M + 1 \tag{65}$$

where $\|\cdot\|_\infty$ is denoting the maximum norm $\|\mathbf{x}\|_\infty \coloneqq \max_i |x_i|$ for a vector $\mathbf{x}$ or its induced matrix norm $\|\mathbf{A}\|_\infty \coloneqq \max_i \sum_j |a_{ij}|$ for a matrix $\mathbf{A}$.

Explicit calculations show that

$$\left\|(\mathbf{1} - \mathbf{D})^{-1}\right\|_\infty \leq \frac{4}{\eta\lambda} \tag{66}$$

under the assumption that $0 \leq \beta < 1$ and $0 < \eta\lambda < 2(1+\beta)$. From here it is straightforward to show that

$$\left\|\left(\mathbf{E} + \mathbf{E}^\top\right)\mathbf{e_1}\right\|_\infty \leq \frac{\tilde{c}}{M(\eta\lambda)^2} \tag{67}$$

where $\tilde{c}$ is a factor of order unity under the constraints $0 \leq \beta < 1$ and $0 < \eta\lambda < 2(1+\beta)$. By substituting this result back into Equation (51) one directly sees that a comparison to the approximation yields

$$\left|1 - \frac{\sigma_\theta^2}{\sigma_{\theta,\text{large}}^2}\right| \leq \frac{c_1}{M(\eta\lambda)^2}\,, \tag{68}$$

$$\left|1 - \frac{\sigma_v^2}{\sigma_{v,\text{large}}^2}\right| \leq \frac{c_2}{M(\eta\lambda)^2}\,, \tag{69}$$

where c_1 and c_2 are again of order unity and the approximation is defined as

$$\begin{aligned}\begin{pmatrix}\sigma^2_{\theta,\text{large}}\\ \sigma^2_{v,\text{large}}\end{pmatrix} &\coloneqq \eta^2\sigma^2_{\delta g}\mathbf{F}\mathbf{e}_1\\ &= \frac{\eta^2\sigma^2_{\delta g}}{(1-\beta)(2(1+\beta)-\eta\lambda)}\cdot\begin{pmatrix}\frac{1+\beta}{\eta\lambda}\\ 2\end{pmatrix}.\end{aligned} \tag{70}$$

Interestingly, one can see that the approximation for large hessian eigenvalues is equivalent to the result we would obtain if we assumed there was no autocorrelation of the noise to begin with.

In the case where the stricter assumption is not true, $M(\eta\lambda)^2 < 1$, but the numerically obtained conditions still hold, $\lambda \gtrsim \lambda_{\text{cross}}$ and $M(1-\beta) \gg 1$, it occurs that $\left\|\left(\mathbf{E}+\mathbf{E}^\top\right)\mathbf{e_1}\right\|_\infty$ is no longer small. But in that case, $\mathbf{F}\left(\mathbf{E}+\mathbf{E}^\top\right)\mathbf{e_1}$ can still be neglected compared to $\mathbf{F}\mathbf{e_1}$, as numerical experiments show.

SMALL HESSIAN EIGENVALUES

To obtain the relations for small hessian eigenvalues, we perform a Taylor expansion with respect to λ with the help of computer algebra. We neglect the terms which are at least of order lambda. Numerical study indicates that these relations hold under the mentioned assumption of $\lambda \lesssim \lambda_{\text{cross}}$ and $M(1-\beta) \gg 1$.

It is straightforward but lengthy to obtain the following expression using the eigendecomposition of $\mathbf{D}$

$$\begin{pmatrix}\sigma^2_\theta\\ \sigma^2_v\end{pmatrix} = \frac{\eta^2\sigma^2_{\delta g}}{2(1-\beta)(1+\beta)}\cdot\begin{pmatrix}\frac{M}{3}\frac{1+\beta}{1-\beta}+\mathcal{O}(\lambda)\\ 2+\mathcal{O}(\lambda)\end{pmatrix} \tag{71}$$

where the zeroth order terms are simplified under approximation $M(1-\beta) \gg 1$.

E.4. Satisfying the assumptions of the correlation time relation

In this section we want to show that the weight and velocity variances resulting from stochastic gradient descent as described above and for Theorem 3 satisfies the necessary assumptions (i) to (iii) of Appendix F such that the velocity correlation time is equal to $\tau_i = 2\sigma^2_{\theta,i}/\sigma^2_{v,i}$. Validity of assumption (i) existence and finiteness of $\mathbf{\Sigma} \coloneqq \text{cov}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k)$, $\mathbf{\Sigma_v} \coloneqq \text{cov}(\mathbf{v}_k, \mathbf{v}_k)$, and $\langle\boldsymbol{\theta}\rangle$ can be inferred from the calculation presented in Appendix E.1. Therefore, we concentrate on assumption (ii) $\lim_{n\to\infty}\text{cov}\left(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+n}\right) = 0$ and (iii) $\lim_{n\to\infty} n\cdot\text{cov}\left(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+n}-\boldsymbol{\theta}_{k+n+1}\right) = 0$. We consider the one dimensional case, but the extension to the multidimensional case is straightforward. Additionally $\langle\boldsymbol{\theta}\rangle = 0$ (see Appendix E.1) and, therefore, the remaining two assumptions (ii) and (iii) can be written as $\lim_{m\to\infty}\langle\theta_k\theta_{k+m}\rangle = 0$ and $\lim_{m\to\infty} m\left(\langle\theta_k\theta_{k+m}\rangle - \langle\theta_k\theta_{k+m+1}\rangle\right) = 0$.

We will now show that for stochastic gradient descent under the assumptions of Theorem 3 the more restrictive relation $\lim_{m\to\infty} m\langle\theta_k\theta_{k+m}\rangle = 0$ is satisfied, from which follows (ii) and (iii). Following Appendix E.2 and using the same notation, we have the relation

$$\langle\mathbf{x}_k\mathbf{x}^\top_{k-m}\rangle = \mathbf{D}\langle\mathbf{x}_{k-1}\mathbf{x}^\top_{k-m}\rangle - \eta\mathbf{e_1}\langle\delta g_k\mathbf{x}^\top_{k-m}\rangle\,, \tag{72}$$

where $\mathbf{x}_k \coloneqq \begin{pmatrix}\theta_k & \theta_{k-1}\end{pmatrix}^\top$. For $m > M$, with M being the number of batches per epoch, the correlation with the noise term on the right hand side of Equation (72) is equal to zero as discussed

in Appendix E.2. By iterating Equation (72), for $m > M$ we have

$$\begin{aligned}\langle \mathbf{x}_k \mathbf{x}_{k-m}^\top \rangle &= \mathbf{D}^{m-M-1} \langle \mathbf{x}_{k-m+M+1} \mathbf{x}_{k-m}^\top \rangle \\ &= \mathbf{D}^{m-M-1} \langle \mathbf{x}_k \mathbf{x}_{k-M-1}^\top \rangle \,. \end{aligned} \tag{73}$$

As described in Appendix E.2, the magnitude of both eigenvalues of $\mathbf{D}$ is strictly smaller than one. This implies that there exists a matrix norm $\|\cdot\|_{\mathbf{D}}$ such that $\|\mathbf{D}\|_{\mathbf{D}} < 1$ from which one can deduce

$$\left\| \langle \mathbf{x}_k \mathbf{x}_{k-m}^\top \rangle \right\|_{\mathbf{D}} \leq \|\mathbf{D}\|_{\mathbf{D}}^{m-M-1} \cdot \left\| \langle \mathbf{x}_k \mathbf{x}_{k-M-1}^\top \rangle \right\|_{\mathbf{D}} \,. \tag{74}$$

Taking the limit of $m \to \infty$ we obtain

$$\begin{aligned}\lim_{m\to\infty} m \left\| \langle \mathbf{x}_k \mathbf{x}_{k-m}^\top \rangle \right\|_{\mathbf{D}} &\leq \text{const} \cdot \lim_{m\to\infty} m \, \|\mathbf{D}\|_{\mathbf{D}}^{m-M-1} \\ &= 0 \,, \end{aligned} \tag{75}$$

and because

$$\langle \mathbf{x}_k \mathbf{x}_{k-m}^\top \rangle = \begin{pmatrix} \langle \theta_k \theta_{k-m} \rangle & \langle \theta_k \theta_{k-m-1} \rangle \\ \langle \theta_{k-1} \theta_{k-m} \rangle & \langle \theta_{k-1} \theta_{k-m-1} \rangle \end{pmatrix} \tag{76}$$

we finally find

$$\begin{aligned} &\lim_{m\to\infty} m \langle \theta_k \theta_{k-m} \rangle = 0 \\ \Rightarrow &\lim_{m\to\infty} m \langle \theta_k \theta_{k+m} \rangle = 0 \,. \end{aligned} \tag{77}$$

Appendix F. Calculation of the Correlation Time Relation

We want to prove the relation

$$\frac{2\sigma_{\theta,i}^2}{\sigma_{v,i}^2} = \frac{\sum_{n=1}^{\infty} n \, \langle v_{k,i} v_{k+n,i} \rangle}{\sum_{n=1}^{\infty} \langle v_{k,i} v_{k+n,i} \rangle} \,. \tag{78}$$

under the following three assumption: (i) Existence and finiteness of $\mathbf{\Sigma} \coloneqq \mathrm{cov}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k)$, $\mathbf{\Sigma}_\mathbf{v} \coloneqq \mathrm{cov}(\mathbf{v}_k, \mathbf{v}_k)$, and $\langle \boldsymbol{\theta} \rangle$. (ii) $\lim_{n\to\infty} \mathrm{cov}\big(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+n}\big) = 0$. (iii) $\lim_{n\to\infty} n \cdot \mathrm{cov}\big(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+n} - \boldsymbol{\theta}_{k+n+1}\big) = 0$. For example, the latter two assumptions hold true if the weight correlation function decays as $\mathrm{cov}\big(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+n}\big) \propto n^{-2}$ or faster.

We assume that that $\langle \boldsymbol{\theta} \rangle$, $\mathbf{\Sigma}_{\boldsymbol{\theta}}$ and $\mathbf{\Sigma}_\mathbf{v}$ exist and are finite. Without loss of generality, let $\langle \boldsymbol{\theta} \rangle = 0$. We consider only the one-dimensional case. For the multidimensional case, simply apply the proof in the direction of any basis vector individually. Note, that the relation still holds if $[\mathbf{\Sigma}, \mathbf{\Sigma}_\mathbf{v}] \neq 0$. In this case, $\sigma_{\theta,i}^2$ and $\sigma_{v,i}^2$ would just be the variances of the weight and the velocity in the given direction but no longer necessarily eigenvalues of $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_\mathbf{v}$.

The remaining two assumptions (ii) and (iii) can now be written as

$$\lim_{m\to\infty} \langle \theta_k \theta_{k+m} \rangle = 0 \tag{79}$$

$$\lim_{m\to\infty} m \big(\langle \theta_k \theta_{k+m} \rangle - \langle \theta_k \theta_{k+m+1} \rangle \big) = 0 \,. \tag{80}$$

We begin the proof with the following chain of equations

$$\begin{aligned}\sigma_\theta^2 &= \langle \theta_k^2 \rangle \\ &= \left\langle (\theta_k - \theta_{k+J} + \theta_{k+J})^2 \right\rangle \\ &= \left\langle (\theta_k - \theta_{k+J})^2 \right\rangle - 2\langle \theta_{k+J}^2 \rangle + 2\langle \theta_k \theta_{k+J} \rangle + \langle \theta_{k+J}^2 \rangle \ , \end{aligned} \tag{81}$$

which holds for any integer J. We have $\langle \theta_{k+J}^2 \rangle = \langle \theta_k^2 \rangle$ since the expectation value cannot depend on k. Additionally, by definition we have $v_k = \theta_k - \theta_{k-1}$ which yields

$$\theta_k - \theta_{k+J} = \sum_{i=1}^{J} v_{k+i} \ . \tag{82}$$

Therefore, we can rewrite Equation (81) as follows

$$\begin{aligned} 2\sigma_\theta^2 &= 2\langle \theta_k \theta_{k+J} \rangle + \sum_{i,j=1}^{J} \langle v_{k+i} v_{k+j} \rangle \\ &= 2\langle \theta_k \theta_{k+J} \rangle + \sum_{i,j=1}^{J} \langle v_k v_{k+j-i} \rangle \\ &= 2\langle \theta_k \theta_{k+J} \rangle + \sum_{m=0}^{J-1} \sum_{n=-m}^{m} \langle v_k v_{k+n} \rangle \ , \end{aligned} \tag{83}$$

where we first shifted the index within the expectation value and then restructured the sum by defining $m \coloneqq \max(i,j) - 1$ and $n \coloneqq j - i$. We now take the limit of $J \to \infty$ and because of Equation (79) and the assumption of a finite σ_θ^2 we have

$$\sum_{m=0}^{\infty} \sum_{n=-m}^{m} \langle v_k v_{k+n} \rangle < \infty \tag{84}$$

$$\Rightarrow \sum_{n=-\infty}^{\infty} \langle v_k v_{k+n} \rangle = 0 \ . \tag{85}$$

We note that $\langle v_k v_{k+n} \rangle = \langle v_k v_{k-n} \rangle$ because we can shift the index, and the two factors commute. Substituting this relation into Equation (85) yields

$$\begin{aligned} \sum_{n=1}^{\infty} \langle v_k v_{k+n} \rangle &= -\frac{1}{2} \langle v_k v_k \rangle \\ &= -\frac{1}{2} \sigma_v^2 \ . \end{aligned} \tag{86}$$

For the second part of the proof we will start again with $v_k = \theta_k - \theta_{k-1}$ and the following sum

$$\begin{aligned} \sum_{n=1}^{m} n \langle v_k v_{k+n} \rangle &= \sum_{n=1}^{m} n \big(2\langle \theta_k \theta_{k+n} \rangle - \langle \theta_{k-1} \theta_{k+n} \rangle - \langle \theta_k \theta_{k+n-1} \rangle \big) \\ &= -\langle \theta_k \theta_k \rangle + \langle \theta_k \theta_{k+m} \rangle + m \big(\langle \theta_k \theta_{k+m} \rangle - \langle \theta_k \theta_{k+m+1} \rangle \big) \ , \end{aligned} \tag{87}$$

where nearly all terms cancel each other again due to the fact that we can shift the index within the expectation value. By taking the limit $m \to \infty$ and using the assumptions (ii) and (iii) (Equations (79) and (80)) we have

$$\sum_{n=1}^{\infty} n\langle v_k v_{k+n}\rangle = -\langle \theta_k \theta_k\rangle \ . \tag{88}$$

Finally, by dividing Equation (88) by Equation (86) we arrive at the final expression

$$\frac{2\sigma_{\theta,i}^2}{\sigma_{v,i}^2} = \frac{\sum_{n=1}^{\infty} n\,\langle v_k v_{k+n}\rangle}{\sum_{n=1}^{\infty}\langle v_k v_{k+n}\rangle} \ . \tag{89}$$

Appendix G. Calculation of the Noise Autocorrelation

We want to calculate the autocorrelation function of epoch-based SGD for a fixed weight vector $\boldsymbol{\theta}$ and under the assumption that the total number of examples is an integer multiple of the number of examples per batch. For that we repeat the following definitions:

$$\boldsymbol{\delta g}_k(\boldsymbol{\theta}) \coloneqq \frac{1}{S}\sum_{n\in\mathcal{B}_k} \boldsymbol{\nabla}\big(l(\boldsymbol{\theta}, x_n) - L(\boldsymbol{\theta})\big) \tag{90}$$

$$\mathcal{B}_k = \{n_1, ..., n_S\} \ldots \text{ batch of step } k\text{, sampeling without replacement within epoch} \tag{91}$$

$$n_j \in \{1, \ldots, N\} \tag{92}$$

$$N \ldots \text{ total number of examples} \tag{93}$$

$$S \ldots \text{ number of examples per batch} \tag{94}$$

We can rewrite the noise terms as follows:

$$\begin{aligned}\boldsymbol{\delta g}_k(\boldsymbol{\theta}) &= \frac{1}{S}\sum_{n\in\mathcal{B}_k} \boldsymbol{\delta g}_{\mathbf{e}}(n, \boldsymbol{\theta}) \\ &= \frac{1}{S}\sum_{n=1}^{N} \boldsymbol{\delta g}_{\mathbf{e}}(n, \boldsymbol{\theta}) s_k^n \end{aligned} \tag{95}$$

$$\begin{aligned} s_k^n &\coloneqq \mathbf{1}_{\mathcal{B}_k}(n) \\ &= \begin{cases} 1 \ \text{ if } \ n \in \mathcal{B}_k \\ 0 \ \text{ if } \ n \notin \mathcal{B}_k \end{cases} \end{aligned} \tag{96}$$

$$\boldsymbol{\delta g}_{\mathbf{e}}(n, \boldsymbol{\theta}) \coloneqq \boldsymbol{\nabla}\big(l(\boldsymbol{\theta}, x_n) - L(\boldsymbol{\theta})\big) \ . \tag{97}$$

Let $h \geq 0$ be fixed. The correlation matrix can be expressed as

$$\begin{aligned}\operatorname{cov}\big(\boldsymbol{\delta g}_k(\boldsymbol{\theta}), \boldsymbol{\delta g}_{k+h}(\boldsymbol{\theta})\big) &= \mathbb{E}\left[\boldsymbol{\delta g}_k(\boldsymbol{\theta})\boldsymbol{\delta g}_{k+h}(\boldsymbol{\theta})^\top\right] \\ &= \frac{1}{S^2}\sum_{n,\tilde{n}=1}^{N} \boldsymbol{\delta g}_{\mathbf{e}}(n, \boldsymbol{\theta})\boldsymbol{\delta g}_{\mathbf{e}}(\tilde{n}, \boldsymbol{\theta})^\top \mathbb{E}\left[s_k^n s_{k+h}^{\tilde{n}}\right] \ . \end{aligned} \tag{98}$$

with $\mathbb{E}[\cdot]$ denoting the limiting average (see Appendix C).

The expectation value of $s_k^n = \mathbf{1}_{\mathcal{B}_k}(n)$ is the probability that example n is part of batch k. Because every example is equally likely to appear in a given batch, this probability is equal to S/N.

$$\begin{aligned}\mathbb{E}\left[s_k^n\right] &= \mathrm{P}(s_k^n = 1) \\ &= \frac{S}{N}\,. \end{aligned} \tag{99}$$

Similarly we can calculate the desired correlation:

$$\begin{aligned}\mathbb{E}\left[s_k^n s_{k+h}^{\tilde{n}}\right] &= \mathrm{P}\big(s_k^n = 1, s_{k+h}^{\tilde{n}} = 1\big) \\ &= \mathrm{P}(s_k^n = 1)\,\mathrm{P}\big(s_{k+h}^{\tilde{n}} = 1 \mid s_k^n = 1\big) \\ &= \frac{S}{N}\,\mathrm{P}\big(s_{k+h}^{\tilde{n}} = 1 \mid s_k^n = 1\big)\,. \end{aligned} \tag{100}$$

The last term can be split up into different probabilities for different values of h. We can also distinguish the case where the two steps k and $k+h$ are within the same epoch ($\mathrm{ep}(k) = \mathrm{ep}(k+h)$) or in different epochs ($\mathrm{ep}(k) \neq \mathrm{ep}(k+h)$).

$$\begin{aligned}\mathrm{P}\big(s_{k+h}^{\tilde{n}} = 1 \mid s_k^n = 1\big) = \delta_{h,0}\cdot \mathrm{P}\big(s_k^{\tilde{n}} = 1 \mid s_k^n = 1\big) + (1-\delta_{h,0})\cdot \\ \Big[\mathrm{P}\big(\mathrm{ep}(k) = \mathrm{ep}(k+h)\big)\mathrm{P}\big(s_{k+h}^{\tilde{n}} = 1 \mid s_k^n = 1, \mathrm{ep}(k) = \mathrm{ep}(k+h)\big) + \\ \mathrm{P}\big(\mathrm{ep}(k) \neq \mathrm{ep}(k+h)\big)\mathrm{P}\big(s_{k+h}^{\tilde{n}} = 1 \mid s_k^n = 1, \mathrm{ep}(k) \neq \mathrm{ep}(k+h)\big)\Big]\,.\end{aligned} \tag{101}$$

The first term of the right hand side of Equation (101) is the probability that a given example occurs in a batch, assuming that we already know one of the examples of that batch.

$$\begin{aligned}\mathrm{P}\big(s_k^{\tilde{n}} = 1 \mid s_k^n = 1\big) &= \mathrm{P}(s_k^n = 1 \mid s_k^n = 1)\cdot\ \delta_{n,\tilde{n}} + \mathrm{P}\big(s_k^{\tilde{n}} = 1 \mid s_k^n = 1, n \neq \tilde{n}\big)\ (1-\delta_{n,\tilde{n}}) \\ &= 1\cdot\ \delta_{n,\tilde{n}} + \frac{S-1}{N-1}\,(1-\delta_{n,\tilde{n}}) \\ &= \frac{N-S}{N-1}\,\delta_{n,\tilde{n}} + \text{const.}\end{aligned} \tag{102}$$

The second term of Equation (101) is multiplied by $(1-\delta_{h,0})$. Therefore, we assume $h \geq 1$ for the following argument. That is, we want to know the probabilities under the assumption that we are comparing examples from different batches. If the two batches are still from the same epoch, examples cannot repeat as the total number of examples is an integer multiple of the number of examples per batch and because of that every example is shown only once per epoch. Therefore, for $h \geq 1$ holds:

$$\begin{aligned}\mathrm{P}\big(s_{k+h}^{\tilde{n}} = 1 \mid s_k^n = 1, \mathrm{ep}(k) = \mathrm{ep}(k+h)\big) &= 0\cdot\delta_{n,\tilde{n}} + \frac{S}{N-1}\,(1-\delta_{n,\tilde{n}}) \\ &= -\frac{S}{N-1}\delta_{n,\tilde{n}} + \text{const.}\end{aligned} \tag{103}$$

If we consider batches from different epochs, the probability becomes independent of the given examples:

$$\begin{aligned}\mathrm{P}\big(s_{k+h}^{\tilde{n}} = 1 \mid s_k^n = 1, \mathrm{ep}(k) \neq \mathrm{ep}(k+h)\big) &= \frac{S}{N} \\ &= \text{const.}\end{aligned} \tag{104}$$

Lastly, we need to know the probability that two given batches k and $k+h$ are from the same epoch:

$$\mathrm{P}\big(\mathrm{ep}(k) = \mathrm{ep}(k+h)\big) = \mathbf{1}_{\{1,\dots,M\}}(h)\,\frac{M-h}{M}\,, \tag{105}$$

where $M = N/S$ is again the number of batches per epoch.

We can now combine all derived probabilities and arrive at the following relation:

$$\begin{aligned}\mathbb{E}\left[s_k^n s_{k+h}^{\tilde{n}}\right] &= \delta_{n,\tilde{n}}\,\frac{S}{N}\frac{N-S}{N-1}\left(\delta_{h,0} - \mathbf{1}_{\{1,\dots,M\}}(h)\frac{S}{N-S}\frac{M-h}{M}\right) + \text{const.}\\ &= \delta_{n,\tilde{n}}\,S^2\left(\frac{1}{S}-\frac{1}{N}\right)\,\frac{1}{N-1}\left(\delta_{h,0} - \mathbf{1}_{\{1,\dots,M\}}(h)\frac{M-h}{M\,(M-1)}\right) + \text{const.}\end{aligned} \tag{106}$$

If we now also consider negative values for h, the expression depends only on the absolute value of h due to symmetry.

By using the following two helpful relations:

$$\begin{aligned}\sum_{n,\tilde{n}=1}^{N} \boldsymbol{\delta}\mathbf{g}_{\mathbf{e}}(n,\boldsymbol{\theta})\boldsymbol{\delta}\mathbf{g}_{\mathbf{e}}(\tilde{n},\boldsymbol{\theta})^{\top}\,\delta_{n,\tilde{n}} &= \sum_{n=1}^{N} \boldsymbol{\delta}\mathbf{g}_{\mathbf{e}}(n,\boldsymbol{\theta})\boldsymbol{\delta}\mathbf{g}_{\mathbf{e}}(n,\boldsymbol{\theta})^{\top}\\ &=: (N-1)\,\mathbf{C}_0(\boldsymbol{\theta})\,,\end{aligned} \tag{107}$$

$$\begin{aligned}\sum_{n,\tilde{n}=1}^{N} \boldsymbol{\delta}\mathbf{g}_{\mathbf{e}}(n,\boldsymbol{\theta})\boldsymbol{\delta}\mathbf{g}_{\mathbf{e}}(\tilde{n},\boldsymbol{\theta})^{\top}\cdot 1 &= \left(\sum_{n=1}^{N}\boldsymbol{\delta}\mathbf{g}_{\mathbf{e}}(n,\boldsymbol{\theta})\right)\left(\sum_{n=1}^{N}\boldsymbol{\delta}\mathbf{g}_{\mathbf{e}}(n,\boldsymbol{\theta})^{\top}\right)\\ &= \boldsymbol{\nabla}\big(L(\boldsymbol{\theta}) - L(\boldsymbol{\theta})\big)\boldsymbol{\nabla}^{\top}\big(L(\boldsymbol{\theta}) - L(\boldsymbol{\theta})\big)\\ &= 0\,,\end{aligned} \tag{108}$$

we can insert the expectation value $\mathbb{E}\left[s_k^n s_{k+h}^{\tilde{n}}\right]$ into Equation (98) and arrive at the final expression:

$$\mathrm{cov}\big[\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta}),\boldsymbol{\delta}\mathbf{g}_{k+h}(\boldsymbol{\theta})\big] = \mathrm{cov}\big[\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta}),\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta})\big]\cdot\left(\delta_{h,0} - \mathbf{1}_{\{1,\dots,M\}}(|h|)\frac{M-|h|}{M(M-1)}\right), \tag{109}$$

$$\mathrm{cov}\big[\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta}),\boldsymbol{\delta}\mathbf{g}_k(\boldsymbol{\theta})\big] = \left(\frac{1}{S}-\frac{1}{N}\right)\mathbf{C}_0(\boldsymbol{\theta})\,. \tag{110}$$

Appendix H. Comparison with principal component analysis

Our approach to analysis sets itself apart from that of Feng & Tu [32] principally in the selection of the basis $\{\mathbf{p_i},, i = 1,\dots,d\}$ used for examining the weights. While they employ the principal components of the weight series - the eigenvectors of $\boldsymbol{\Sigma}$ - we use the eigenvectors of the hessian matrix $\mathbf{H}(\boldsymbol{\theta}_\mathbf{K})$ computed at the beginning of the analysis period.

This choice enables us to directly create plots of variances and correlation time against the hessian eigenvalue for each corresponding direction. Feng & Tu devised a landscape-dependent flatness parameter F_i for every direction $\mathbf{p_i}$. However, with the assistance of the second derivative $F_i \approx \left(\partial^2 L(\boldsymbol{\theta})/\partial\theta_i^2\right)^{-\frac{1}{2}}$, where $\theta_i = \boldsymbol{\theta}\cdot\mathbf{p_i}$, this parameter can be approximated, provided this second derivative retains a sufficiently positive value. Hence, in the eigenbasis of the hessian matrix,

the flatness parameter can be approximated as $F_i \approx \lambda_i^{-\frac{1}{2}}$, facilitating comparability between our analysis and that of Feng & Tu.

The principal component basis, as used by Feng & Tu, holds a distinct advantage. For our analysis, we needed to eliminate the near-linear trajectory of the weights by deducting the mean velocity. However, in Feng & Tu's analysis, this movement is automatically subsumed in the first principal component due to its pronounced variance. Hence, there's no necessity for additional subtraction of this drift in the weight covariance eigenbasis.

Yet, the weight covariance eigenbasis has a significant shortcoming: it yields artifacts. This is because $\mathbf{\Sigma}$ is calculated as an average over a finite data set, skewing its eigenvalues from the anticipated distribution. Consequently, the resultant eigenvectors may not align perfectly with the expected ones. This issue is further exacerbated due to the high dimensionality of the underlying space.

The artifact issue becomes evident in Figure 4, which displays synthetic data generated through stochastic gradient descent within an isotropic quadratic potential coupled with isotropic noise. With 2,500 dimensions, the model mirrors the scale of a layer in the fully connected neural network that Feng & Tu investigated. The weight series comprises 12,000 steps, which correspond to ten epochs of training this network. Analyzing this data with the weight covariance eigenbasis seemingly suggests anisotropic variance and correlation time. However, if the data is inspected without any basis change, both the variance and correlation time appear isotropically distributed as anticipated.

To navigate around this key issue associated with the eigenbasis of $\mathbf{\Sigma}$, we adopted the eigenvectors of the Hessian matrix. Unlike $\mathbf{\Sigma}$, the Hessian is not computed as an average over update steps but can, in theory, be precisely calculated for any given weight vector. Consequently, the Hessian matrix does not suffer from finite size effects. The difference between these two bases for actual data is visible in Figure 5. Here, we analyzed only the weights of the first convolutional layer of the LeNet from the main text to ensure comparability with Feng & Tu's results. In this specific comparison, the network was trained without weight decay. Due to this and the fact that we are only investigating the weights of one layer, λ_{cross} is significantly larger than all Hessian eigenvalues. As a result, when analyzing in the eigenbasis of the Hessian matrix related to this layer, both the variance and the correlation time align well with the prediction for smaller Hessian eigenvalues.

However, analyzing in the eigenbasis of the weight covariance matrix, the correlation time appears heavily dependent on the second derivative of the loss in the given direction. Additionally, the relationship between the weight variance and the second derivative shifts and more closely aligns with Feng & Tu's results as the power law exponent is significantly larger than one. The first principal component, which Feng & Tu referred to as the drift mode, stands out due to its unusually long correlation time. This is to be expected, as this is the direction in which the weights are moving at an approximately constant velocity.

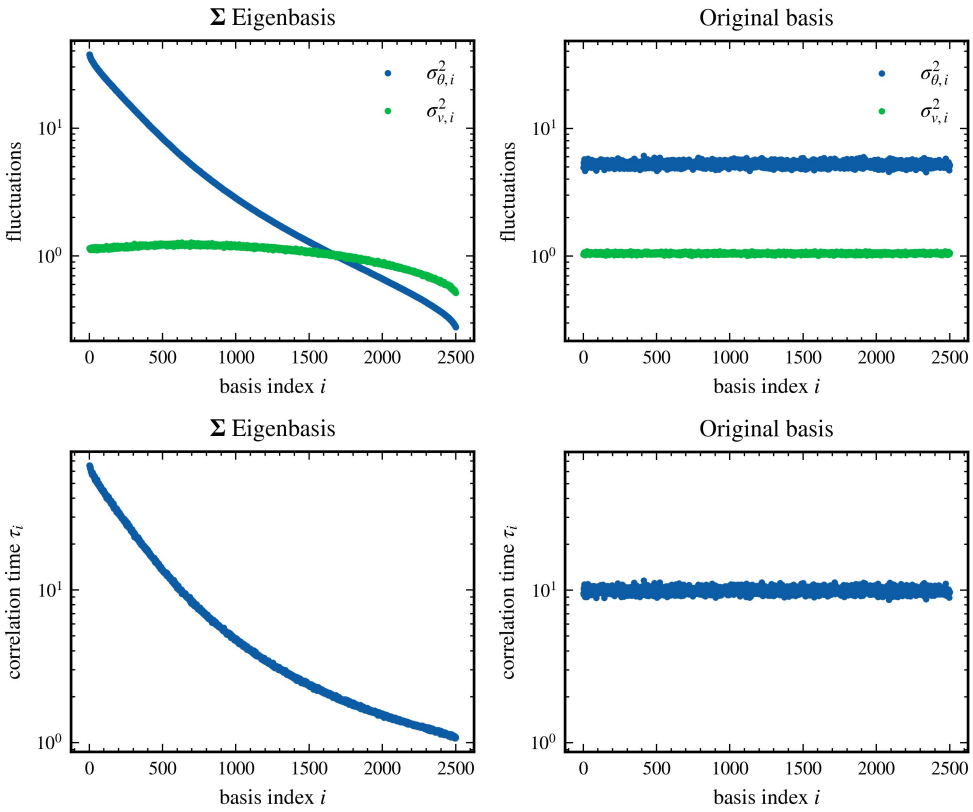

Figure 4: Comparison of weight and velocity fluctuations for synthetic data analyzed in two different bases. We define the variance in weight, $\sigma^2_{\theta,i}$, as $\mathbf{p_i}^\top \mathbf{\Sigma} \mathbf{p_i}$, and the variance in velocity, $\sigma^2_{v,i}$, as $\mathbf{p_i}^\top \mathbf{\Sigma_v} \mathbf{p_i}$. The correlation time, τ_i, is given by $2\sigma^2_{\theta,i}/\sigma^2_{v,i}$. The synthetic data was generated by simulating SGD for 12,000 steps within a 2,500-dimensional space featuring an isotropic quadratic potential and isotropic noise. In the original basis analysis, both the variance and correlation time, as expected, retain isotropy. However, when the analysis is conducted in the eigenbasis of the weight covariance matrix, a pronounced anisotropy emerges.

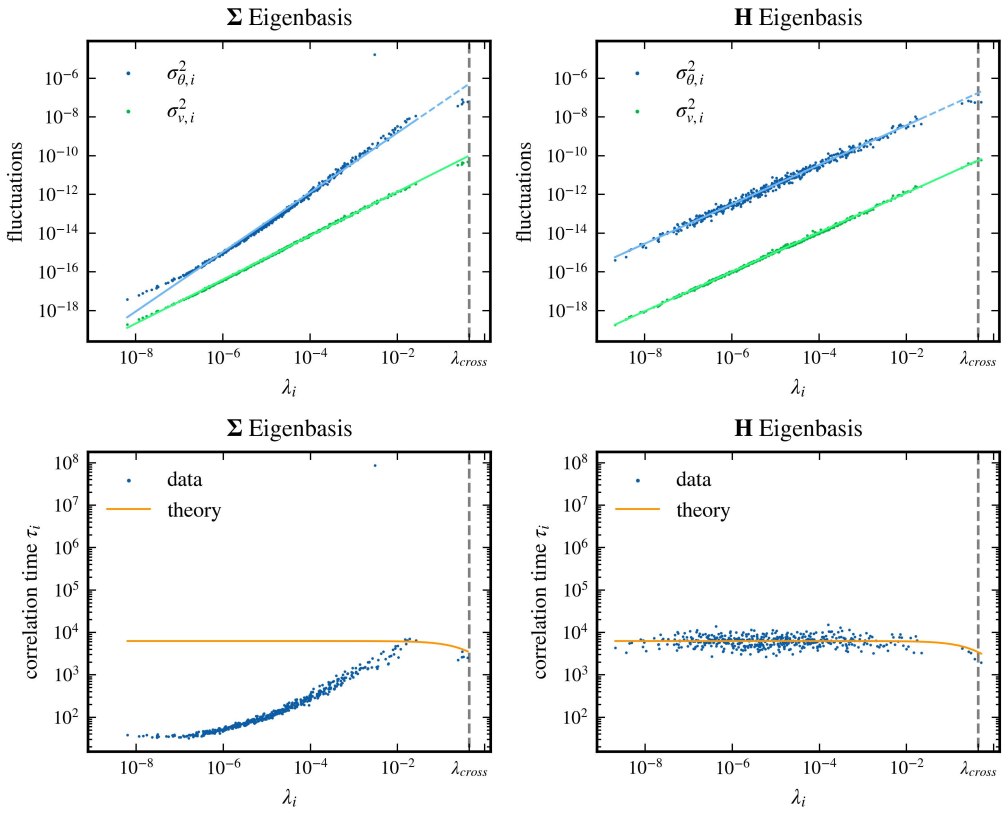

Figure 5: Comparison of weight and velocity variances for all 450 weights of the first convolutional layer of the LeNet, as discussed in the main text, analyzed in two different bases. In order to facilitate a more directly comparable analysis to Feng & Tu [32], the network was trained without weight decay for this specific analysis and the analysis period was limited to 10 epochs, as opposed to the usual 20 epochs. The columns represent different bases: for the left column $\mathbf{p_i}$ are the eigenvectors of $\mathbf{\Sigma}$ and for the right column $\mathbf{p_i}$ are the eigenvectors of $\mathbf{H}$. The mean velocity was subtracted in the right column. The rows illustrate the weight and velocity variance, $\sigma^2_{\theta,i} = \mathbf{p_i}^\top \mathbf{\Sigma} \mathbf{p_i}$, $\sigma^2_{v,i} = \mathbf{p_i}^\top \mathbf{\Sigma_v} \mathbf{p_i}$ (top row), and the correlation time $\tau_i = 2\sigma^2_{\theta,i}/\sigma^2_{v,i}$ (bottom row). The second derivative of the corresponding direction is depicted on the x-axis, $\lambda_i = \mathbf{p_i}^\top \mathbf{H} \mathbf{p_i}$. The top row solid lines indicate fit regions for a linear fit. For the $\mathbf{H}$ eigenbasis, the respective exponent of the power law relation is 1.018 ± 0.008 for weight variance and 1.017 ± 0.002 for velocity variance with a 2σ-error. For the $\mathbf{\Sigma}$ eigenbasis, the corresponding exponent is 1.537 ± 0.012 for weight variance and 1.134 ± 0.002 for velocity variance.

Appendix I. Drawing with replacement

To confirm that the results obtained are indeed affected by the correlations present in SGD noise, due to the epoch-based learning strategy, we reapply the analysis described in the main text. In this instance, however, we deviate from our previous method of choosing examples for each batch within an epoch without replacement. Instead, we select examples with replacement from the complete pool of examples for every batch. This modification during the analysis period allows a more complete assessment of the impact of correlations on the derived results.

Figure 6 offers clear visual proof that when examples are selected with replacement, the previously noted anti-correlations within the SGD noise vanish. This observation confirms our hypothesis that the anti-correlations mentioned in the main text are indeed an outcome of the epoch-based learning technique. Consequently, we can predict that this change will influence the behaviour of the weight and velocity variance. As previously discussed, the theoretical results we have achieved for Hessian eigenvectors with eigenvalues exceeding λ_{cross} conform to what one would predict in the absence of any correlation within the noise. Therefore, when examples are drawn with replacement, we anticipate the weight variance to be isotropic in all directions, while the velocity variance should remain unchanged.

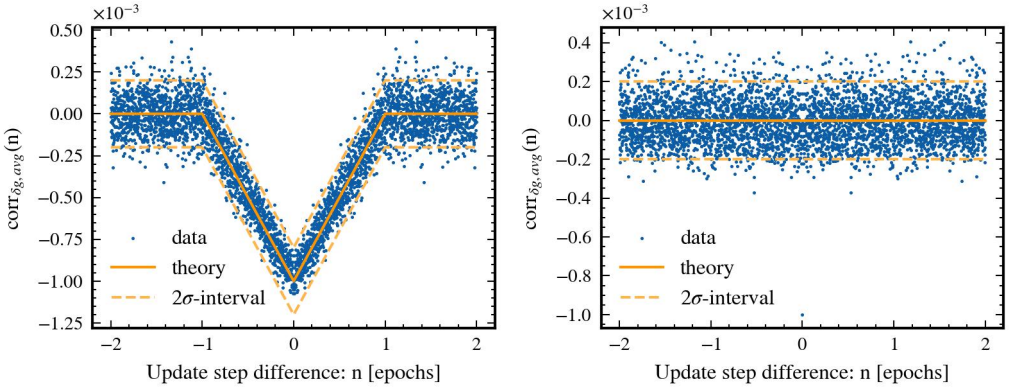

Figure 6: Autocorrelations of the SGD noise compared for drawing examples without replacement (left) and with replacement (right).

Upon reviewing Figure 7, it is clear that the velocity variance stays unchanged as predicted. However, while the weight variance remains constant for a broader subset of Hessian eigenvalues, it reduces for extremely small eigenvalues. Likewise, the correlation time is still limited for these minuscule Hessian eigenvalues. These deviations can be attributed to the finite time frame of the analysis period, comprising 20,000 update steps. This limited time window sets a cap on the maximum correlation time, consequently leading to a decreased weight variance for these small Hessian eigenvalues. Despite this, it is noteworthy that this maximum correlation time is still roughly one order of magnitude longer than the maximum correlation time induced by the correlations arising from the epoch-based learning approach.

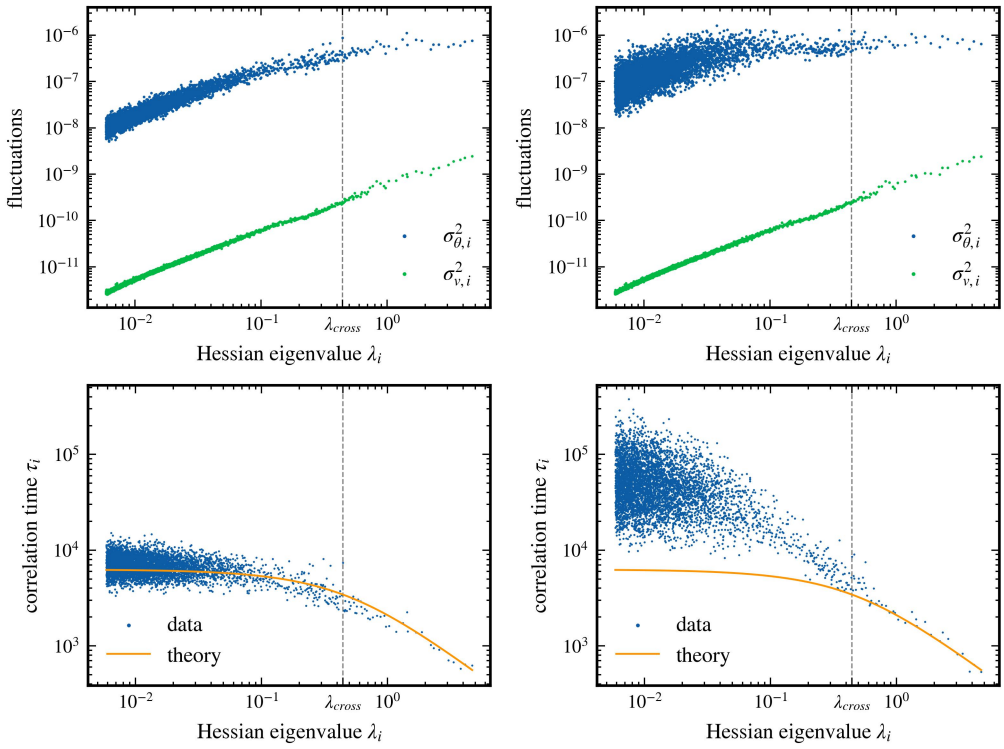

Figure 7: Relationship between Hessian eigenvalues and the variances of weights and velocities, as well as correlation times. For the left column the examples are drawn in epochs without replacement and for the right column the examples are drawn with replacement.

Appendix J. Loss and Accuracy during Training

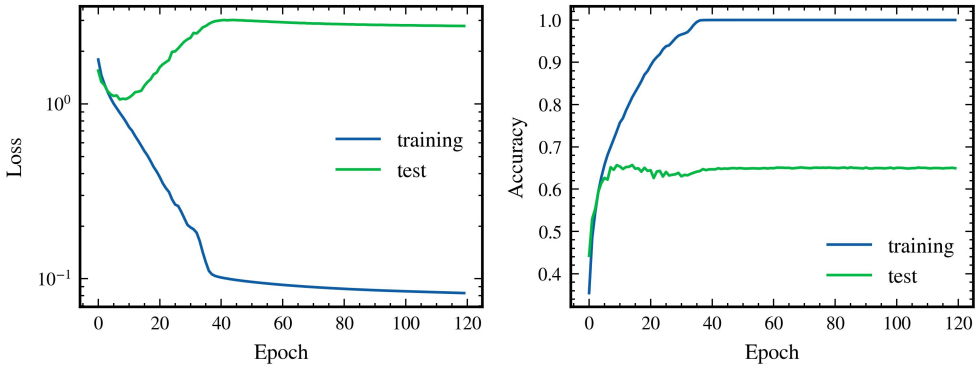

Figure 8: The evolution of the loss (left) and accuracy (right) during training of LeNet described in the main text. The statistics are shown for both training and test set. For the first 100 epochs, the exponential learning rate decay was used, and for the last 20 epochs, the learning rate was fixed at the final value of the exponential decay.

Appendix K. Testing different Hyperparameters

In this section we examine the dependence of the theoretical predictions on the three hyperparameters learning rate η, momentum β and batch size S. For this, we train the LeNet again for 100 epochs, using an exponential learning rate schedule that reduces the learning rate by a factor of 0.98 every epoch and afterwards we perform the numerical analysis as described in the main text.

However, we now train the network several times, always varying one of the hyperparameters while keeping the other two fixed. If not varied, the momentum was set to 0.90 and the batch size was set to 64. To ensure that training is always successful and 100% training accuracy is achieved, the initial learning rate was set to 0.005 when the batch size is varied and to 0.02 when the momentum is varied. Five different values are examined for each hyperparameter. To investigate the dependencies on the learning rate, the values 0.005, 0.01, 0.02, 0.03, and 0.04 were used for training. For momentum, the values 0.00, 0.50, 0.75, 0.90, and 0.95 were examined, and for batch size, the values 32, 50, 64, 100, and 128 were examined.

In addition, the training was repeated for five different seeds for each hyperparameter combination in order to obtain reliable results. This results in a relatively high computational cost. To reduce this, for the analysis in this section the weight and velocity variances are examined only in the subspace of the 2,000 largest Hessian eigenvalues and associated eigenvectors.

Figure 9 shows as an example the weight and velocity variances as well as the correlation times for different values of the batch size for one training seed each. It can be seen that the theory is not only valid for the hyperparameter combination from the previous section, but is also generally applicable for different hyperparameters. In particular, we see that there is still good agreement with the theory even if the strictly necessary condition for the theoretical derivation of the noise autocorrelation, that the number of batches per epoch $M = N/S$ is an integer, is not met.

To further examine the predictions of the theory for the hyperparametric dependencies, we now focus on the two quantities of the maximum correlation time τ_{SGD} and the Hessian eigenvalue crossover value λ_{cross} and recall the theoretical predictions for these quantities:

$$\tau_{\mathrm{SGD}} = \frac{N}{3S}\frac{1+\beta}{1-\beta}\,, \tag{111a}$$

$$\lambda_{\mathrm{cross}} = \frac{3S(1-\beta)}{\eta N}\,, \tag{111b}$$

where N is the number of examples in the training data set.

For the evaluation of the dependence of these variables on the hyperparameters, they were determined as follows for the various hyperparameter combinations using the data from the respective correlation time plot. For the maximum correlation time τ_{SGD}, the average of all correlation times was taken for which the corresponding Hessian eigenvalue is smaller than the theoretical crossover value. However, the result for the numerically determined maximum correlation time is not significantly different when simply taking the average of all determined correlation times for a hyperparameter combination, since only very few Hessian eigenvalues are larger than the crossover value.

For the numerical determination of the crossover value λ_{cross}, a linear function was first fitted to the correlation times of the 20 largest Hessian eigenvalues in the log-log plot of the correlation times against the Hessian eigenvalues. In this region of the first 20 values, the correlation time is always clearly dependent on the Hessian eigenvalue and does not yet belong to the region of constant

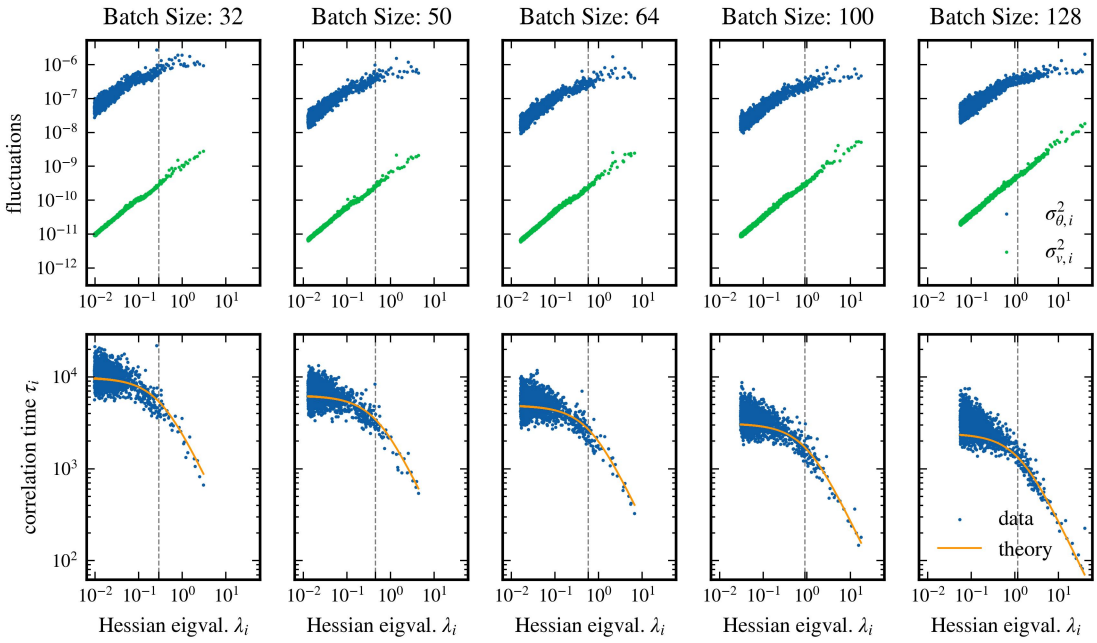

Figure 9: Testing the LeNet training with different hyperparameters. Here the relationship between the Hessian eigenvalues and the variances and correlation times for varying batch size is shown as an example. The momentum was set to 0.90 and the initial learning rate was set to 0.005.

correlation times. The intersection of the fitted line with the numerically determined maximum correlation time τ_{SGD} is then taken as the crossover value λ_{cross}. If the numerically determined correlation times follow the theory exactly, then the correlation times determined in this way for τ_{SGD} and λ_{cross} would also follow the theory accurately.

And indeed, Figure 10 shows a good agreement between the theory and the numerically determined values, although it should be noted that the deviations are larger than the random fluctuations between the different seeds.

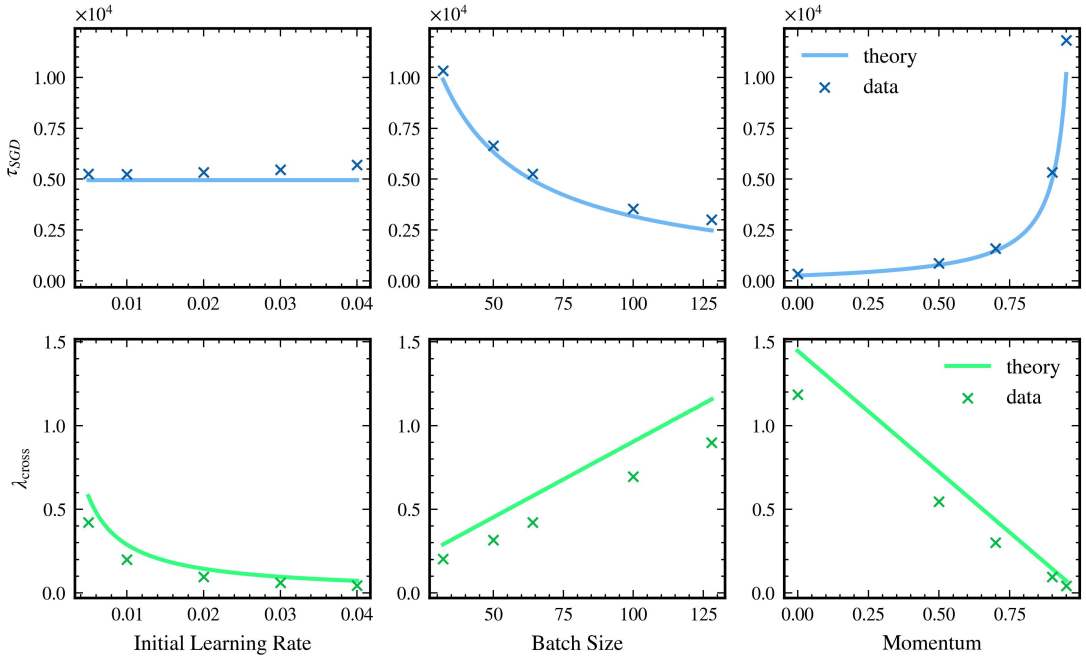

Figure 10: The empirically determined maximum correlation time, τ_{SGD}, and the empirically determined crossover value λ_{cross} for the training of the LeNet with different hyperparameters, averaged over five different seeds for each set of hyperparameters. The predictions of our theory are shown as a solid line. The fluctuations between different seeds are smaller than the marker size and are therefore not included in the figure.

Appendix L. Different Network Architecture

To further confirm our theoretical predictions within trained networks, in this section we turn to a more modern architecture. Instead of the previously used LeNet network, we examine the ResNet-20 network [5]. It is a convolutional network with significantly more convolutional layers than LeNet. It also uses residual blocks with residual connections, which allows for deeper network structures. As our loss function, we again employed Cross Entropy, along with an L2 regularization with a prefactor of 10^{-4} and we did not use batch normalization. The number of layers, which is already indicated in the name with 20, is significantly higher than in the LeNet with just five layers. With approximately 272,000 parameters, the ResNet-20 also has significantly more parameters and the computational cost is significantly higher.

Therefore, in this section we limit ourselves to examining the weight and velocity variances in the subspace of the 400 largest Hessian eigenvalues and associated eigenvectors. The network was trained with SGD for 100 epochs using the same exponential learning rate schedule as before, with a learning rate of $5 \cdot 10^{-3}$, a momentum parameter of 0.9, and a minibatch size of 50. This setup achieves 100% training accuracy and 73% testing accuracy. In Figure 11 one can observe a good agreement between the theory predictions for the variances and correlation times and the numerical observations.

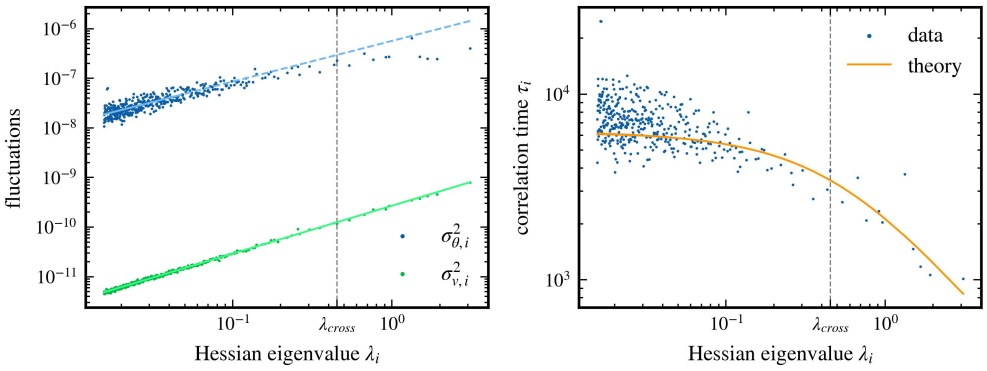

Figure 11: Relationship between Hessian eigenvalues and the variances and correlation times for a ResNet-20 trained on CIFAR10. The mean velocity of the weight trajectory was subtracted. The solid lines in the left panel indicate the regions utilized for a linear fit. The exponents resulting from the power law relationship are 0.820 ± 0.099 for weight variance and 0.965 ± 0.005 for velocity variance, with a 2σ-error. The analysis was performed for the 400 largest Hessian eigenvalues and corresponding eigendirections.

Appendix M. Effects of Non-Commutativity

The exact variance equation of Theorem 3 and the calculation shown in Appendix E are to some extent still valid even if the previously mentioned assumption $[\mathbf{C}, \mathbf{H}] \neq 0$ is not given. In particular, the result for the correlation time holds independently of the commutativity assumption. However, the calculated weight and velocity variances are no longer eigenvalues of the corresponding covariance matrices, but only the variances in the directions of the chosen eigenvector of the Hessian matrix.

Assuming that $\mathbf{C}$ and $\mathbf{H}$ do not necessarily commute, we can still project the update equations onto an arbitrary Hessian eigenvector $\mathbf{p}_i$ with eigenvalue λ_i, which gives us

$$v_{k,i} = -\eta\lambda_i\theta_{k-1,i} + \beta v_{k,i} - \eta \mathbf{p}_i \cdot \boldsymbol{\delta}\mathbf{g}_k \,, \tag{112a}$$

$$\theta_{k,i} = (1-\eta\lambda_i)\theta_{k-1,i} + \beta v_{k,i} - \eta \mathbf{p}_i \cdot \boldsymbol{\delta}\mathbf{g}_k \,. \tag{112b}$$

Since $\mathbf{p}_i \cdot \boldsymbol{\delta}\mathbf{g}_k$ is independent of weights and velocity, these two equations are decoupled for each individual Hessian eigenvector. As $\mathbf{p}_i \cdot \boldsymbol{\delta}\mathbf{g}_k$ still follows the proposed anti-correlation, the calculations of Appendix E can be performed similarly. Therefore, the theory prediction for the correlation time τ_i, defined as the ratio between the weight and the velocity variance in the eigendirection $\mathbf{p}_i$ of the Hessian, is still valid. However, the weight and the velocity variance in the eigendirection $\mathbf{p}_i$ are no longer eigenvalues of the covariance matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_\mathbf{v}$ if $\mathbf{p}_i$ is not also an eigenvector of $\mathbf{C}$.

To better understand the consequences of a situation where $\mathbf{C}$ and $\mathbf{H}$ do not commute, we perform an SGD simulation in an artificial quadratic loss landscape where we introduce epoch-based noise with anti-correlations as described in Theorem 1. However, we choose the Hessian matrix $\mathbf{H}$ of the loss and the gradient noise covariance matrix $\mathbf{C}$ to be non-commuting. The results are shown in Figure 12.

When $\mathbf{H}$ and $\mathbf{C}$ are approximately proportional to each other, to an extent reported in the literature [31], the results still show good agreement with our theory and there are only small deviations

for very small Hessian eigenvalues. If there is no similarity between $\mathbf{H}$ and $\mathbf{C}$, then the results for the variances no longer follow our theory, but the prediction for the correlation time τ_i still shows good agreement with the numerical data, as expected.

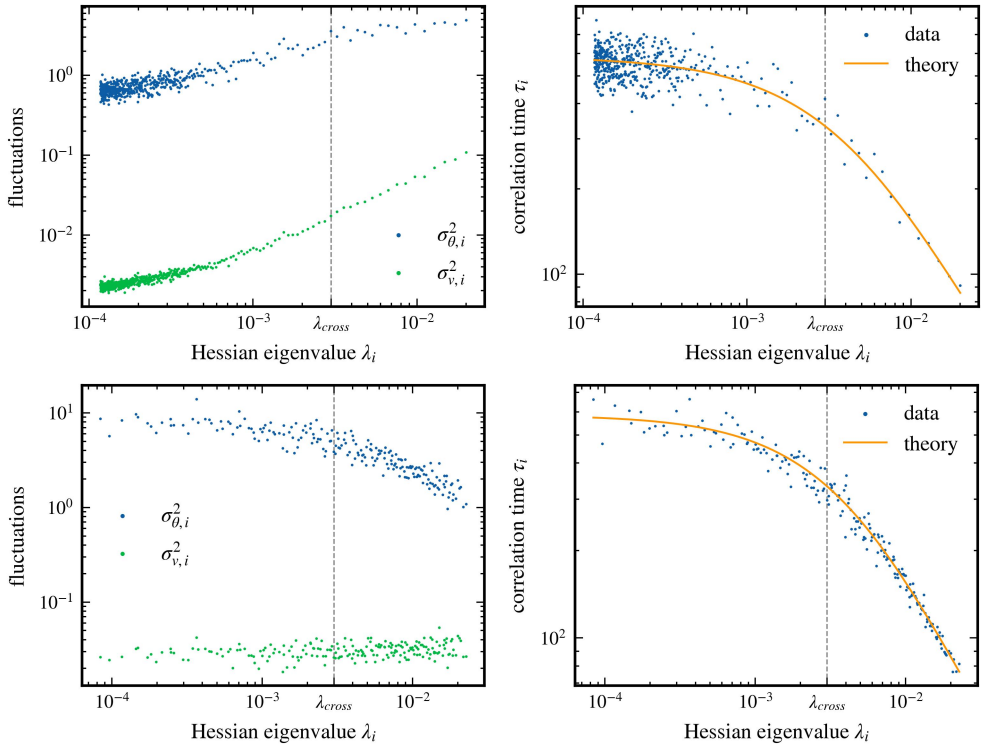

Figure 12: Variances and correlation times of an SGD trajectory in an artificial quadratic potential with epoch-based noise. The quantities are examined in the Hessian eigenbasis and plotted against the corresponding eigenvalue. For the top two panels, the covariance matrix of the noise $\mathbf{C}$ is equal to the Hessian $\mathbf{H}$ plus a random matrix $\mathbf{M}$ of small magnitude ($\mathbf{M} = \frac{1}{d}\mathbf{X}\mathbf{X}^\top$ with $\mathbf{X} \in \mathbb{R}^{d\times d}$ and $X_{ij} \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 0.02$, $d = 500$). The cosine similarity between $\mathbf{C}$ and $\mathbf{H}$ is 0.96 which is a value that has been found empirically in real networks before [31]. Here, our theory for the variances still holds except below a certain small eigenvalue. For the bottom two panels, the Hessian and the covariance matrix are two independent random matrices. There is no longer a relationship between the Hessian eigenvalue and the velocity covariance. However, the theoretical prediction for the correlation time still holds.