Mini-batch Noise Lowers Sharpness via Dominant-Subspace Fluctuations

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

During SGD training, the gradients often align strongly with the dominant subspace spanned by the top-k eigenvectors of the Hessian of the loss. While this seems to naturally imply that loss reduction mainly occurs within this space, prior work has shown that updates within this dominant subspace make no meaningful progress in reducing the loss. In this work, we argue that the dominant subspace is better understood not as the main space for loss reduction, but as a key subspace for explaining the sharpness dynamics of mini-batch SGD. To explain the role of the dominant subspace in reducing top-k sharpness, we show how the averaged gradient over fluctuations in the dominant directions produces a sharpness correction term, and derive a sharpness correction term induced by mini-batch noise in the dominant directions. Experimental results show that adding the derived correction term to GD brings the sharpness evolution of GD closer to that of SGD.

1. Introduction

Understanding the dynamics of training deep neural networks is one of the main topics of machine learning. During stochastic gradient descent (SGD) training, it has been observed that the Hessian of the loss often has a small number of large outlier eigenvalues [9, 24–27], and training gradients are known to align strongly with the *dominant subspace*, which is spanned by the top-k Hessian eigenvectors [9, 10]. This suggests that the effective training dynamics of SGD may be low-dimensional, and it naturally leads to the expectation that loss reduction mainly occurs within this space.

However, prior work [29] reports results that contradict this interpretation. Specifically, when the SGD update is projected onto the dominant subspace, training achieves no meaningful loss reduction. In contrast, when the update is projected onto the *bulk subspace*, which is the orthogonal complement of the dominant subspace, the loss decreases similarly to standard SGD. Does the dominant subspace therefore contribute nothing to training?

In this paper, we argue that the main role of the dominant subspace lies in reducing the top-k sharpness of the loss landscape. Specifically, we observe that the dominant-projected update significantly reduces sharpness even though it has little impact on loss reduction. Then, through experiments involving controlled perturbations along different subspaces (*dominant* and *random*), we show that only perturbations in the dominant directions reduce sharpness. To explain this effect, we first show that averaging the gradient over fluctuations in the dominant directions yields a deterministic sharpness correction term. We then derive the deterministic correction term induced by mini-batch noise in the dominant directions. Finally, we empirically show that adding this correction term to GD produces sharpness dynamics similar to SGD.

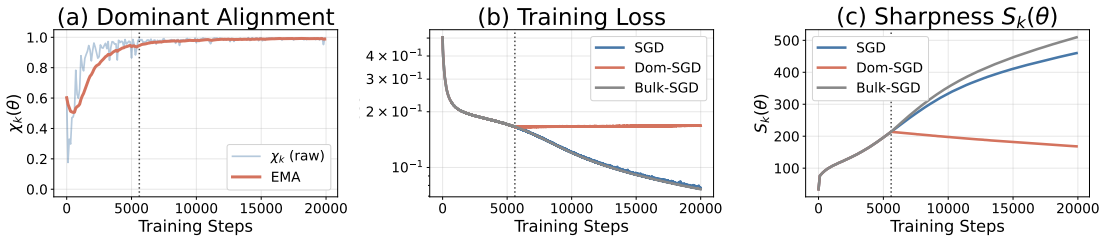

Figure 1: **The dominant component does not reduce loss but reduces sharpness.** (a) Training gradients align with the dominant subspace. (b) Dom-SGD fails to reduce loss, while Bulk-SGD continues to train. (c) Dom-SGD lowers S_k, whereas Bulk-SGD maintains higher S_k than SGD.

2. Setup and Motivation

Setup. Let $L : \mathbb{R}^d \to \mathbb{R}$ be the loss function. For a mini-batch B with mini-batch loss L_B, define the mini-batch noise as $\xi_B(\theta) := \nabla L_B(\theta) - \nabla L(\theta)$, where $\mathbb{E}_B[\xi_B(\theta) \mid \theta] = 0$. We write mini-batch SGD as $\theta_{t+1} = \theta_t - \eta(\nabla L(\theta_t) + \xi_{B_t}(\theta_t))$, and full-batch GD as $\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t)$. Let $(\lambda_i(\theta), e_i(\theta))$ be the eigenpairs of $H(\theta) = \nabla^2 L(\theta)$, ordered so that $\lambda_1(\theta) \geq \cdots \geq \lambda_d(\theta)$. We define the *dominant subspace* as the space spanned by the top-k eigenvectors, $E_{\text{dom}}(\theta) = \text{span}\{e_1(\theta), \ldots, e_k(\theta)\}$, and define its orthogonal complement $E_{\text{bulk}}(\theta) = E_{\text{dom}}(\theta)^{\perp}$ as the *bulk subspace*. We also define the projections onto these spaces as the dominant projection $P_{\text{dom}}(\theta)$ and the bulk projection $P_{\text{bulk}}(\theta) = I - P_{\text{dom}}(\theta)$. We define top-$k$ sharpness as $S_k(\theta) = \sum_{j=1}^{k} \lambda_j(\theta)$. We restrict our focus to the stable learning-rate regime, where $\lambda_1(\theta) < 2/\eta$ is maintained during training.

2.1. Motivation and Main Observation

Let us define the dominant alignment metric $\chi_k(\theta) = \|P_{\text{dom}}(\theta)\nabla L(\theta)\|/\|\nabla L(\theta)\|$, which quantifies the relative magnitude of the training-loss gradient lying in the dominant subspace along the SGD trajectory. As shown in Figure 1(a), the SGD training gradient is strongly concentrated in the dominant subspace from early in training, and this tendency becomes stronger at later stages [9, 10]. This observation suggests that SGD dynamics are closely related to the dominant subspace.

However, as shown in Figure 1(b), Dom-SGD, which projects the SGD update onto the dominant subspace, does not substantially reduce the training loss. Instead, Bulk-SGD, which keeps only the bulk component, reduces the loss as effectively as SGD [29]. Prior work [29, 33] explains this phenomenon using the river-valley intuition. In this view, as shown in Figure 2, the dominant directions correspond to the high-curvature valley walls, where updates do not reduce the loss, whereas the bulk directions follow the valley floor, where loss reduction mainly occurs.

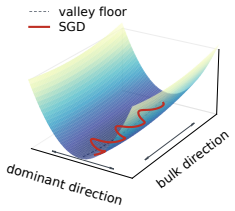

Figure 2: River-valley intuition.

Nevertheless, prior work has reported that models trained with Bulk-SGD alone fail to fully recover the final accuracy gains achieved by SGD [36]. This suggests that the dominant subspace, while not primarily responsible for loss reduction, may still play a meaningful role in training. Its contribution may instead lie in aspects of the optimization dynamics that are not directly captured by loss reduction alone.

In this work, we argue that the dominant subspace plays an important role in reducing the top-k sharpness of the loss landscape. As shown in Figure 1(c), Bulk-SGD follows almost the same loss curve as SGD, but its top-k sharpness becomes higher than that of SGD. In contrast, Dom-SGD

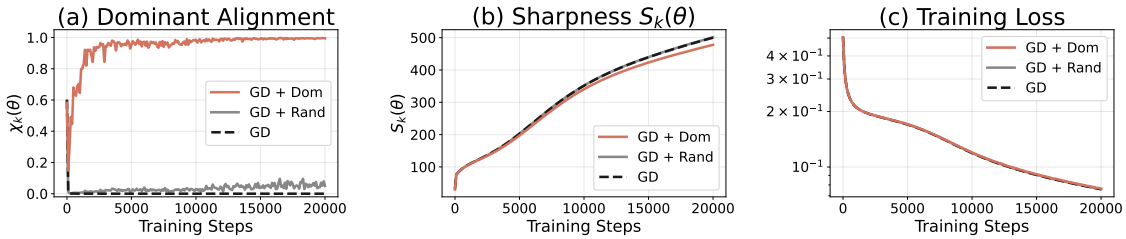

Figure 3: **Dominant perturbations reduce top-k sharpness.** We compare GD, GD with dominant mean-zero perturbations (GD+Dom), and GD with random mean-zero perturbations (GD+Rand). The two perturbation runs are matched in rank and noise scale ($k = 10, \rho = 0.1$). (a) Dominant perturbations induce strong dominant alignment. (b) Only dominant perturbations reduce S_k. (c) Both perturbed runs leave the GD loss curve nearly unchanged. All experiments are run on an MLP.

keeps the loss high while substantially reducing top-k sharpness, suggesting that the dominant subspace is connected to sharpness reduction rather than loss reduction. In the next section, we present experiments examining whether motion along the dominant directions is in fact the factor that reduces top-k sharpness.

3. Dominant-Subspace Fluctuations Reduce Sharpness

The previous observation showed that the dominant subspace is related to sharpness reduction rather than loss reduction. However, this simple comparison between Dom-SGD and Bulk-SGD does not reveal whether this sharpness reduction comes from the **gradient component in the dominant subspace** or from **stochastic motion within that subspace**.

To separate these effects, we conduct a controlled noise injection experiment that keeps the mean update equal to full-batch GD while adding only mean-zero perturbations in different subspaces. Specifically, we compare the perturbed GD dynamics $\theta_{t+1} = \theta_t - \eta(\nabla L(\theta_t) + \zeta_t)$ with ζ_t sampled from different subspaces. For the dominant perturbation, we set $\zeta_t \sim \mathcal{N}(0, \rho^2 P_{\text{dom}}(\theta_t))$, and for the random perturbation, we set $\zeta_t \sim \mathcal{N}(0, \rho^2 P_{\text{rand}})$, where P_{rand} is a random k-dimensional orthogonal projection fixed at initialization.

First, as shown in Figure 3(c), GD, GD with random perturbations (GD+Rand), and GD with dominant perturbations (GD+Dom) follow almost the same training-loss curve, indicating that these perturbations do not significantly alter the rate of loss reduction. However, as shown in Figure 3(a) and 3(b), their trajectories exhibit substantial geometric differences: GD and GD+Rand maintain low χ_k and high S_k, whereas GD+Dom drives χ_k close to 1 and maintains a lower top-k sharpness S_k than either GD or GD+Rand.

These results show two things. First, as suggested by the previous observation, alignment with the dominant subspace does not imply that loss reduction occurs along the dominant directions. Second, and more importantly, stochastic motion alone within the dominant subspace can actually affect sharpness dynamics while barely changing the loss curve. Together, these results indicate that the role of the dominant subspace lies not in providing a direction for loss reduction, but rather in changing the **local geometry associated with the dominant directions**, especially top-k sharpness. In the next section, we show that this effect arises from the interaction between the covariance of a mean-zero displacement and local variations of the Hessian.

4. Sharpness Correction Term

In this section, we show that when the parameter fluctuates in the dominant directions around a reference point, averaging the gradient leaves a ***sharpness correction term***. To show this, we consider a displacement $\delta \in E_{\text{dom}}(\theta_c)$ from a reference point θ_c, with $\mathbb{E}[\delta] = 0$, and write the nearby parameter as $\theta_c + \delta$. Expanding $\nabla L(\theta_c + \delta)$ using a Taylor expansion around θ_c, we get:

$$\nabla L(\theta_c + \delta) = \nabla L(\theta_c) + H(\theta_c)\delta + \frac{1}{2}\nabla(\delta^\top H\delta)(\theta_c) + O(\|\delta\|^3). \tag{1}$$

The first term in Eq. (1) is the gradient at the reference point, the second term is the linear term in the displacement, and the third term is a second-order term that captures the local variation of the Hessian. Averaging this over δ gives:

$$\mathbb{E}_\delta[\nabla L(\theta_c + \delta)] = \nabla L(\theta_c) + \underbrace{H(\theta_c)\mathbb{E}[\delta]}_{=0} + \frac{1}{2}\nabla \operatorname{Tr}(H(\theta_c)C) + O(\mathbb{E}\|\delta\|^3), \qquad C := \mathbb{E}[\delta\delta^\top]. \tag{2}$$

Here, C is held fixed when taking the derivative. As shown in Eq. (2), the averaged gradient is the sum of the gradient at the reference point and an additional term $\frac{1}{2}\nabla \operatorname{Tr}(H(\theta_c)C)$. This term is determined by the displacement covariance C and acts to lower the covariance-weighted curvature represented by $\operatorname{Tr}(HC)$. When C is supported in the dominant subspace, it lowers the weighted curvature associated with top-k sharpness, so we refer to it as a *sharpness correction term*.

Mini-batch-induced displacement covariance. Finally, we derive the displacement covariance induced by mini-batch noise in the local recursion and substitute it into the sharpness correction term. This shows that the sharpness effect of mini-batch noise can be represented as a deterministic correction term.

To define the local displacement created by mini-batch noise around a reference point θ_c, we compare the mini-batch SGD update with the full-batch GD update. Consider a local state $\theta_c + \delta_s$, where $\delta_s \in E_{\text{dom}}(\theta_c)$, and let $P_c := P_{\text{dom}}(\theta_c)$. The next displacement δ_{s+1} is defined as

$$\delta_{s+1} := P_c(\theta^+_{\text{mb}} - \theta^+_{\text{gd}}), \qquad \theta^+_{\text{mb}} := \theta_c + \delta_s - \eta\nabla L_{B_s}(\theta_c + \delta_s), \qquad \theta^+_{\text{gd}} := \theta_c - \eta\nabla L(\theta_c).$$

We then linearize the mini-batch gradient at $\theta_c + \delta_s$ around θ_c:

$$\nabla L_{B_s}(\theta_c + \delta_s) \approx \nabla L(\theta_c) + H_c\delta_s + \xi_{B_s}(\theta_c), \qquad H_c := H(\theta_c).$$

Substituting this into the definition of δ_{s+1} gives

$$\delta_{s+1} = P_c\big(\delta_s - \eta H_c\delta_s - \eta\xi_{B_s}(\theta_c)\big) = A_c\delta_s - \eta P_c\xi_{B_s}(\theta_c), \qquad A_c := P_c(I - \eta H_c)P_c. \tag{3}$$

Taking second moments in Eq. (3) gives

$$C_{s+1} = A_cC_sA_c^\top + \eta^2\Sigma_{\text{dom}}(\theta_c), \qquad (C_s := \mathbb{E}[\delta_s\delta_s^\top]), \quad (\Sigma_{\text{dom}}(\theta_c) := P_c \operatorname{Cov}(\xi_B(\theta_c))P_c).$$

Here, Σ_{dom} denotes the mini-batch noise covariance projected onto the dominant subspace. Let C_{mb} be the stationary limit of the covariance sequence C_s. Then $C_{\text{mb}} = A_cC_{\text{mb}}A_c^\top + \eta^2\Sigma_{\text{dom}}$. Therefore, substituting C_{mb} into Eq. (2) gives the following:

$$\mathbb{E}[\nabla L(\theta_c + \delta)] \approx \nabla L(\theta_c) + \frac{1}{2}\nabla \operatorname{Tr}(H(\theta_c)C_{\text{mb}}), \qquad C_{\text{mb}} = \eta^2\sum_{\ell=0}^{\infty} A_c^\ell\Sigma_{\text{dom}}(A_c^\top)^\ell.$$

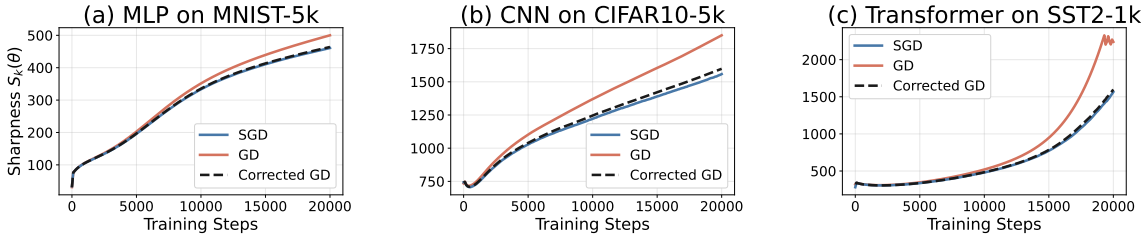

Figure 4: **The sharpness correction term brings GD closer to mini-batch SGD in top-k sharpness.** Across MLP, CNN, and Transformer, full-batch GD reaches higher top-k sharpness S_k, whereas adding the derived sharpness correction term to GD brings its top-k sharpness curve closer to that of mini-batch SGD.

Thus, C_{mb} is not merely the mini-batch noise covariance, but the displacement covariance produced by the local recursion, and therefore it also depends on the local Hessian structure and the learning rate. With this covariance, the effect of mini-batch noise on the averaged gradient is represented as a deterministic correction term (see Appendix C for the detailed derivation).

5. Experiments

In this section, we empirically test whether the sharpness correction term derived in Section 4 reproduces the low sharpness of mini-batch SGD. For the experiments, we compare the sharpness dynamics of full-batch GD, mini-batch SGD, and corrected GD across three model architectures: MLP, CNN, and Transformer [31]. Here, corrected GD is obtained by adding our sharpness correction term to GD, $\theta_{t+1} = \theta_t - \eta\nabla L(\theta_t) + \eta b_{\text{corr}}(\theta_t)$, where $b_{\text{corr}}(\theta) := -\frac{1}{2}\nabla_\vartheta \operatorname{Tr}(H(\vartheta)C_{\text{mb}}(\theta))\big|_{\vartheta=\theta}$. Detailed experimental settings are given in Appendix D.

Figure 4 shows that full-batch GD moves toward higher S_k than mini-batch SGD in all three models, whereas corrected GD stays much closer to the low sharpness level of SGD. In particular, for the MLP, corrected GD almost overlaps with SGD, and for the CNN and Transformer, it also keeps S_k lower than GD and substantially reduces the *sharpness gap* with SGD.

This result shows that the sharpness correction term we derived in Section 4 explains much of the sharpness dynamics of SGD, empirically showing that the dominant subspace helps explain the lower top-k sharpness of mini-batch SGD, rather than directly providing a direction for reducing the loss.

6. Conclusion

Prior work by Song et al. [29] shows that the dominant subspace, the top-k eigenspace of the loss Hessian, does not provide meaningful directions for loss reduction, leaving its explanatory role unclear. In this paper, we show that this dominant subspace plays an important role in reducing top-k sharpness under SGD. Specifically, we show that perturbations within the dominant subspace reduce top-k sharpness, and derive the sharpness correction term induced by mini-batch noise. Empirically, we show that adding this correction term to GD brings its sharpness evolution closer to that of mini-batch SGD, demonstrating that the dominant subspace matters not as a direction for loss reduction, but as the subspace through which mini-batch noise lowers sharpness.

References

[1] Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 639–668. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/andriushchenko22a.html.

[2] Peter L. Bartlett, Philip M. Long, and Olivier Bousquet. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *Journal of Machine Learning Research*, 24(316):1–36, 2023. URL http://jmlr.org/papers/v24/23-043.html.

[3] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jh-rTtvkGeM.

[4] Jeremy Cohen, Alex Damian, Ameet Talwalkar, J Zico Kolter, and Jason D. Lee. Understanding optimization in deep learning with central flows. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sIE2rI3ZPs.

[5] Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability, 2023. URL https://arxiv.org/abs/2209.15594.

[6] Shenyang Deng, Boyao Liao, Zhuoli Ouyang, Tianyu Pang, Minhak Song, and Yaoqing Yang. Suspicious alignment of sgd: A fine-grained step size condition analysis, 2026. URL https://arxiv.org/abs/2601.11789.

[7] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1019–1028. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/dinh17b.html.

[8] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

[9] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2232–2241. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ghorbani19b.html.

[10] Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace, 2018. URL https://arxiv.org/abs/1812.04754.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 01 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

[12] Stanis law Jastrzebski, Zac Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Amos Storkey, and Yoshua Bengio. Three factors influencing minima in SGD, 2018. URL https://openreview.net/forum?id=rJma2bZCW.

[13] Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

[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. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

[15] Andrew V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2): 517–541, 2001. doi: 10.1137/S1064827500366124. URL https://doi.org/10.1137/S1064827500366124.

[16] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[17] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5905–5914. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kwon21b.html.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[19] Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2101–2110. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/li17f.html.

[20] Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *Journal of Machine Learning Research*, 20(40):1–47, 2019. URL http://jmlr.org/papers/v20/17-526.html.

[21] Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling SGD with stochastic differential equations (SDEs). In A. Beygelzimer, Y. Dauphin, P. Liang, and

J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=goEdyJ_nVQI.

[22] Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss? –a mathematical framework. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=siCt4xZn5Ve.

[23] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference, 2018. URL https://arxiv.org/abs/1704.04289.

[24] Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5012–5021. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/papyan19a.html.

[25] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020. URL http://jmlr.org/papers/v21/20-933.html.

[26] Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond, 2017. URL https://openreview.net/forum?id=B186cP9gx.

[27] Levent Sagun, Utku Evci, V. Ugur Güney, Yann N. Dauphin, and Léon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *CoRR*, abs/1706.04454, 2017. URL http://arxiv.org/abs/1706.04454.

[28] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

[29] Minhak Song, Kwangjun Ahn, and Chulhee Yun. Does SGD really happen in tiny subspaces? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=v6iLQBoIJw.

[30] 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 Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3503–3513. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/thomas20a.html.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=5spDgWmpY6x.

[33] Kaiyue Wen, Zhiyuan Li, Jason S. Wang, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=m51BgoqvbP.

[34] Yeming Wen, Kevin Luk, Maxime Gazeau, Guodong Zhang, Harris Chan, and Jimmy Ba. An empirical study of stochastic gradient descent with structured covariance noise. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3621–3631. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/wen20a.html.

[35] Lei Wu, Mingze Wang, and Weijie Su. The alignment property of sgd noise and how it helps select flat minima: A stability analysis. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 4680–4693. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/1e55c38dd7d465c2526ae29d7ec85861-Paper-Conference.pdf.

[36] Daniyar Zakarin and Sidak Pal Singh. Accelerating neural network training along sharp and flat directions, 2025. URL https://arxiv.org/abs/2505.11972.

[37] 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 Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7654–7663. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhu19e.html.

Appendix

Appendix A. Related Work

Hessian spectra and dominant alignment. The training-loss Hessian of a neural network often has a structure that separates into a small number of large outlier eigenvalues and a broad bulk spectrum [9, 24–27]. It has also been repeatedly observed that, during training, the training-loss gradient is strongly aligned with a low-dimensional subspace spanned by a small number of eigenvectors corresponding to the large eigenvalues of the Hessian [9, 10]. These observations suggest that SGD dynamics in a high-dimensional parameter space are closely related to low-dimensional structure.

Projected-update experiments. Song et al. [29] conducted projected-update experiments to test whether SGD can still reduce the training loss when its updates are restricted to the top Hessian eigenspace, referred to as the dominant subspace. Surprisingly, Dom-SGD, which trains using only the SGD update projected onto the dominant subspace, barely reduces the training loss. In contrast, Bulk-SGD, which trains using the SGD update with its dominant component removed, continues to reduce the loss similarly to standard SGD. To address this phenomenon, Song et al. [29] interpret the loss landscape as an ill-conditioned valley. In this view, the dominant directions correspond to high-curvature valley walls, so Dom-SGD mostly moves across the walls and has difficulty reducing the loss, whereas Bulk-SGD follows the valley floor and continues to train. This intuition is also similar to the river-valley loss landscape [33]. Deng et al. [6] further analyze projected SGD updates by deriving step-size conditions for loss decrease. However, these loss-based explanations account for why a dominant-projected update has difficulty making loss progress, but not why the dominant subspace is connected to lower top-k sharpness. This question is the starting point of our work.

Sharpness and flatness. Sharpness and flatness have long been studied as loss-landscape geometry for explaining generalization. Early work suggested that flat minima, where the loss does not change much under small perturbations around the solution, are associated with better generalization [11]. Later, it was also observed that large-batch training can converge to sharper minimizers than small-batch training and show a generalization gap, with worse test performance despite similar training loss [14]. This view also led to optimization methods that directly search for flat minima. A representative method is SAM, and since then, related variants and theoretical analyses have also been actively studied [1, 2, 8, 17, 32]. On the other hand, since sharpness-based measures can be sensitive to parameterization or experimental settings, there is also a view that sharpness alone has limits in explaining generalization [7, 13].

SGD noise and implicit regularization. SGD noise has been studied as a source of implicit regularization related to flatness and sharpness. Prior work has studied SGD noise through the stationary distribution of constant-step SGD [23], or approximated SGD using SDEs or stochastic modified equations [19–21]. Along with this view, other studies explored how the scale and covariance structure of SGD noise affect optimization and generalization [12, 30, 34, 37]. In particular, some analyses suggest that this covariance structure is related to sharp directions and affects flat-minima selection [35]. There is also work showing that mini-batch SGD noise near a zero-loss manifold induces a drift toward lower sharpness of the loss [22]. Like this line of work, we study how SGD noise affects sharpness, but we focus on the displacement covariance induced in the dominant subspace rather than on the mini-batch noise covariance itself.

Edge of stability. In large-step-size GD, the largest Hessian eigenvalue is observed to increase up to around the stability threshold $2/\eta$, after which training continues without diverging, although the loss

becomes non-monotonic. This phenomenon is called the Edge of Stability (EoS) [3]. Subsequent theoretical analyses showed that motion caused by instability in the top Hessian direction can lower sharpness through higher-order terms [5], and that the time-averaged trajectory of large-step GD near EoS can be described by a deterministic flow [4]. Our mechanism is closely related to these analyses in that motion in high-curvature directions leaves a higher-order correction after averaging. However, the source of the correction is different. While these works study oscillations that arise in large-step GD, we focus on fluctuations induced by mini-batch noise in SGD.

Appendix B. Notation and Assumptions

In this section, we fix the notation and assumptions used in Appendix C.

B.1. Notation

Optimization setup. We use $L : \mathbb{R}^d \to \mathbb{R}$ to denote the training loss. For a mini-batch B, we write L_B for the corresponding mini-batch loss.

We write

$$H(\theta) := \nabla^2 L(\theta)$$

for the Hessian matrix at θ. For a mini-batch B, we define the mini-batch noise by

$$\xi_B(\theta) := \nabla L_B(\theta) - \nabla L(\theta), \qquad \mathbb{E}_B[\xi_B(\theta) \mid \theta] = 0.$$

We denote the corresponding noise covariance by

$$\Sigma_\xi(\theta) := \mathrm{Cov}_B(\xi_B(\theta) \mid \theta).$$

Mini-batch SGD with learning rate $\eta > 0$ is written as

$$\theta_{t+1} = \theta_t - \eta \nabla L_{B_t}(\theta_t) = \theta_t - \eta\big(\nabla L(\theta_t) + \xi_{B_t}(\theta_t)\big).$$

Dominant Hessian subspace. We denote the eigenpairs of $H(\theta)$ by $(\lambda_i(\theta), e_i(\theta))_{i=1}^d$, ordered as

$$\lambda_1(\theta) \geq \lambda_2(\theta) \geq \cdots \geq \lambda_d(\theta).$$

We define the dominant subspace as

$$E_{\mathrm{dom}}(\theta) := \mathrm{span}\{e_1(\theta), \ldots, e_k(\theta)\},$$

and write $P_{\mathrm{dom}}(\theta)$ for the orthogonal projection onto $E_{\mathrm{dom}}(\theta)$. We also write

$$E_{\mathrm{bulk}}(\theta) := E_{\mathrm{dom}}(\theta)^\perp, \qquad P_{\mathrm{bulk}}(\theta) := I - P_{\mathrm{dom}}(\theta).$$

We define the top-k sharpness as

$$S_k(\theta) := \sum_{i=1}^{k} \lambda_i(\theta).$$

Local displacements. For a reference point θ_c, we write a nearby parameter as

$$\theta_c + \delta,$$

where δ denotes the local displacement. In the local averaging calculations, δ is mean-zero, and we write

$$C(\theta_c) = \mathbb{E}[\delta\delta^\top]$$

for its displacement covariance. When the displacement is supported on $E_{\mathrm{dom}}(\theta_c)$, the covariance C is supported on the same subspace.

Local linear map. For a reference point θ_c, we define

$$A_c := P_{\mathrm{dom}}(\theta_c)\big(I - \eta H(\theta_c)\big)P_{\mathrm{dom}}(\theta_c),$$

viewed as a linear map on $E_{\mathrm{dom}}(\theta_c)$. This is the linear part of the local recursion derived in Appendix C.3.

B.2. Local Assumptions

We use the following local assumptions at each reference point θ_c.

Assumption 1 (Local Taylor regularity). Around θ_c, the loss L is locally C^4. This is used to apply the Taylor expansion of the gradient and to control the local remainder terms.

Assumption 2 (Eigengap). When derivatives of S_k or P_{dom} are used, we assume

$$\lambda_k(\theta_c) > \lambda_{k+1}(\theta_c).$$

This ensures that the top-k eigenspace is separated from the rest of the spectrum near θ_c.

Assumption 3 (Moments and mini-batch noise). We use the following moment and noise assumptions.

(i) Local displacement moments. The local displacements used in the averaging calculations satisfy

$$\mathbb{E}[\delta] = 0, \qquad \mathbb{E}\|\delta\|^3 < \infty.$$

In the local recursion, we initialize with

$$\mathbb{E}[\delta_0] = 0.$$

(ii) Mini-batch noise. The mini-batch noise satisfies

$$\mathbb{E}_B[\xi_B(\theta_c) \mid \theta_c] = 0, \qquad \|\Sigma_\xi(\theta_c)\| < \infty.$$

(iii) Local noise freezing. For the local recursion, we write the noise variation as

$$\xi_{B_s}(\theta_c + \delta_s) = \xi_{B_s}(\theta_c) + r_{\xi,s},$$

and ignore $r_{\xi,s}$ when linearizing the mini-batch gradient around θ_c.

Assumption 4 (Local stability). When taking the stationary limit of the covariance recursion, we assume that the selected dominant eigenvalues satisfy

$$0 < \lambda_k(\theta_c) \le \lambda_1(\theta_c) < \frac{2}{\eta}.$$

Since the local linear map A_c has eigenvalues $1 - \eta\lambda_i(\theta_c)$ on $E_{\mathrm{dom}}(\theta_c)$, this implies

$$\rho(A_c) < 1.$$

Appendix C. Derivation of the Sharpness Correction Term

In this section, we derive in more detail the sharpness correction term presented briefly in the main text. Section 4 showed that when the local displacement lies in the dominant directions and has zero mean, averaging the gradient leaves a sharpness correction term, and described how this correction is induced by mini-batch noise. Here, we give the same calculation in more detail and show how the dominant perturbation experiment in Section 3 and mini-batch noise each lead to a sharpness correction term through their displacement covariance. The notation and assumptions follow Appendix B.

First, in Appendix C.1, we show that averaging $\nabla L(\theta_c + \delta)$ over a local displacement with zero mean leaves a covariance-weighted curvature term: the sharpness correction term. Next, in Appendix C.2, we specialize this sharpness correction term to the dominant perturbation experiment in Section 3, and show why it lowers top-k sharpness in that case. Finally, in Appendix C.3, we compute the displacement covariance induced by mini-batch noise through a local recursion, and show how it determines the mini-batch-induced sharpness correction term.

C.1. Sharpness correction from local displacement

We compute the covariance-weighted curvature term that remains when we average the gradient around a reference point. To do this, fix a reference point θ_c, and let δ be a local displacement around θ_c with $\mathbb{E}[\delta] = 0$. We write the nearby parameter as $\theta_c + \delta$, and define the displacement covariance by

$$C(\theta_c) := \mathbb{E}[\delta\delta^\top].$$

Motivated by the dominant perturbation experiment in Section 3, we consider the case where the displacement around θ_c lies in the dominant subspace. That is, we take $\delta \in E_{\mathrm{dom}}(\theta_c)$, and so we view C as a covariance supported on $E_{\mathrm{dom}}(\theta_c)$.

Now expand the gradient at $\theta_c + \delta$ around θ_c. The second-order Taylor expansion of the gradient is

$$\nabla L(\theta_c + \delta) = \nabla L(\theta_c) + H(\theta_c)\delta + \frac{1}{2}\nabla(\delta^\top H\delta)(\theta_c) + R_3(\theta_c, \delta). \tag{4}$$

In Eq. (4), the first term is the gradient at the reference point, and the second term is the linear term in the displacement. The third term is a second-order term that captures the local variation of the Hessian around θ_c, and $R_3(\theta_c, \delta)$ is the higher-order remainder.

Now average Eq. (4) over δ. Since $\mathbb{E}[\delta] = 0$, the linear term vanishes after averaging:

$$\mathbb{E}_\delta[H(\theta_c)\delta] = H(\theta_c)\mathbb{E}_\delta[\delta] = 0.$$

After this cancellation, the only nontrivial term left to average is the second-order term containing the Hessian variation. To compute this, let ϑ be a temporary variable and write

$$\delta^\top H(\vartheta)\delta = \mathrm{Tr}\big(H(\vartheta)\delta\delta^\top\big)\,.$$

Then, since $C(\theta_c) = \mathbb{E}[\delta\delta^\top]$,

$$\mathbb{E}_\delta[\delta^\top H(\vartheta)\delta] = \mathrm{Tr}(H(\vartheta)C(\theta_c))\,.$$

Here, $C(\theta_c)$ is held fixed when taking the derivative with respect to ϑ. Therefore,

$$\mathbb{E}_\delta\left[\nabla(\delta^\top H\delta)(\theta_c)\right] = \nabla_\vartheta\,\mathrm{Tr}(H(\vartheta)C(\theta_c))|_{\vartheta=\theta_c}\,.$$

Thus, the averaged gradient is

$$\mathbb{E}_\delta[\nabla L(\theta_c+\delta)] = \nabla L(\theta_c) + \frac{1}{2}\,\nabla_\vartheta\,\mathrm{Tr}(H(\vartheta)C(\theta_c))|_{\vartheta=\theta_c} + \mathcal{R}_3(\theta_c), \tag{5}$$

where

$$\mathcal{R}_3(\theta_c) := \mathbb{E}_\delta[R_3(\theta_c,\delta)].$$

Under Assumptions 1 and 3 in Appendix B.2, this remainder is $O(\mathbb{E}_\delta\|\delta\|^3)$.

As shown in Eq. (5), averaging the gradient around the reference point adds a covariance-weighted curvature term to the gradient at the reference point. We refer to this additional term as a sharpness correction term. This term is determined once the covariance C is specified. We first consider the covariance that corresponds to the dominant perturbation experiment in Section 3, and then derive the covariance created by mini-batch noise.

C.2. Sharpness correction from dominant perturbations

For the dominant perturbation experiment in Section 3, we specify the covariance that enters the sharpness correction term in Eq. (5). We then show that the resulting correction lowers top-k sharpness.

A dominant perturbation can be viewed as the case where the covariance is isotropic inside the dominant subspace. That is, for some scalar variance $\sigma^2 > 0$, let

$$C = \sigma^2 P_{\mathrm{dom}}(\theta_c) \tag{6}$$

We substitute the covariance in Eq. (6) into the correction term in Eq. (5). Under Assumption 2 in Appendix B.2, S_k is locally differentiable, and the standard eigenvalue derivative formula gives

$$\nabla_\vartheta\,\mathrm{Tr}(H(\vartheta)P_{\mathrm{dom}}(\theta_c))|_{\vartheta=\theta_c} = \nabla S_k(\theta_c).$$

Therefore, by Eq. (6),

$$\nabla_\vartheta\,\mathrm{Tr}(H(\vartheta)C)|_{\vartheta=\theta_c} = \sigma^2\nabla S_k(\theta_c). \tag{7}$$

Substituting this into Eq. (5) gives

$$\mathbb{E}_\delta[\nabla L(\theta_c+\delta)] = \nabla L(\theta_c) + \frac{\sigma^2}{2}\nabla S_k(\theta_c) + \mathcal{R}_3(\theta_c). \tag{8}$$

As shown in Eq. (8), when the covariance is isotropic inside the dominant subspace, the averaged gradient has the form of the gradient at the reference point plus the term $\frac{\sigma^2}{2}\nabla S_k(\theta_c)$. Thus, in the negative-gradient update, the term $-\frac{\sigma^2}{2}\nabla S_k(\theta_c)$ is added on top of the full-gradient term, and this term acts in the direction that lowers S_k. This explains why the dominant perturbation in Section 3 shows lower top-k sharpness than GD.

C.3. Sharpness correction from mini-batch noise

In Appendix C.2, we considered the case where the displacement covariance can be specified directly, as in the dominant perturbation experiment in Section 3. For mini-batch noise, this covariance is not directly given. We now derive it from a local recursion around a reference point.

Local recursion. Fix a reference point θ_c. To model the local displacement created by mini-batch noise around this point, we introduce a local recursion around θ_c, indexed by s. At local step s, we write the local state as $\theta_c + \delta_s$, with $\delta_s \in E_{\mathrm{dom}}(\theta_c)$, and draw a fresh mini-batch B_s.

To define one step of this local recursion, we take the difference between the mini-batch update from $\theta_c + \delta_s$ and the full-batch GD update from θ_c, and project it onto the dominant subspace. First, define the mini-batch update at $\theta_c + \delta_s$ and the full-batch GD update at θ_c as

$$\theta^+_{\mathrm{mb}} := \theta_c + \delta_s - \eta \nabla L_{B_s}(\theta_c + \delta_s), \tag{9}$$

$$\theta^+_{\mathrm{gd}} := \theta_c - \eta \nabla L(\theta_c) \tag{10}$$

respectively. Then the next displacement in the local recursion is defined as

$$\delta_{s+1} := P_{\mathrm{dom}}(\theta_c)(\theta^+_{\mathrm{mb}} - \theta^+_{\mathrm{gd}}). \tag{11}$$

That is, after subtracting the full-batch GD step from θ_c, we project the remaining local difference onto the dominant subspace and use the result as δ_{s+1}.

Next, we approximate the mini-batch gradient $\nabla L_{B_s}(\theta_c + \delta_s)$ in Eq. (9) around the reference point θ_c. By the definition of mini-batch noise, $\xi_B(\theta) := \nabla L_B(\theta) - \nabla L(\theta)$, we have

$$\nabla L_{B_s}(\theta_c + \delta_s) = \nabla L(\theta_c + \delta_s) + \xi_{B_s}(\theta_c + \delta_s). \tag{12}$$

Now approximate the two terms in Eq. (12) around θ_c as follows:

$$\nabla L(\theta_c + \delta_s) = \nabla L(\theta_c) + H(\theta_c)\delta_s + O(\|\delta_s\|^2), \qquad \xi_{B_s}(\theta_c + \delta_s) \approx \xi_{B_s}(\theta_c).$$

Therefore, after substituting this local approximation into Eq. (12) and dropping the higher-order term, we get

$$\nabla L_{B_s}(\theta_c + \delta_s) \approx \nabla L(\theta_c) + H(\theta_c)\delta_s + \xi_{B_s}(\theta_c). \tag{13}$$

Eq. (13) shows that the mini-batch gradient at $\theta_c + \delta_s$ is approximated by the full-gradient term at θ_c, the linear response to the local displacement, and the mini-batch noise term at θ_c.

Now use Eq. (13) to compute δ_{s+1}. By Eq. (9) and Eq. (11),

$$\begin{aligned}
\delta_{s+1} &\approx P_{\mathrm{dom}}(\theta_c)\Big[\theta_c + \delta_s - \eta\big(\nabla L(\theta_c) + H(\theta_c)\delta_s + \xi_{B_s}(\theta_c)\big) - \big(\theta_c - \eta \nabla L(\theta_c)\big)\Big] \\
&= P_{\mathrm{dom}}(\theta_c)\big(\delta_s - \eta H(\theta_c)\delta_s - \eta \xi_{B_s}(\theta_c)\big) \\
&= A_c \delta_s - \eta P_{\mathrm{dom}}(\theta_c)\xi_{B_s}(\theta_c).
\end{aligned} \tag{14}$$

Eq. (14) is the local recursion around the reference point θ_c. Here, $A_c\delta_s$ is the term by which the previous local displacement is carried forward through A_c, and $-\eta P_{\mathrm{dom}}(\theta_c)\xi_{B_s}(\theta_c)$ is the term by which the current mini-batch noise is newly injected into the dominant subspace.

Displacement covariance recursion. Eq. (14) gives the recursion for the local displacement δ_s. Now we compute the local displacement covariance induced by this recursion. First, define the s-th local displacement covariance as

$$C_s := \mathbb{E}[\delta_s \delta_s^\top]. \tag{15}$$

Next, define the dominant noise covariance at the reference point θ_c as

$$\Sigma_{\mathrm{dom}}(\theta_c) := P_{\mathrm{dom}}(\theta_c)\Sigma_\xi(\theta_c)P_{\mathrm{dom}}(\theta_c). \tag{16}$$

Since each B_s is a mini-batch drawn freshly at the local step, $\xi_{B_s}(\theta_c)$ is independent of δ_s, and $\mathbb{E}[\xi_{B_s}(\theta_c) \mid \theta_c] = 0$. From these conditions, the cross terms satisfy

$$\mathbb{E}[\delta_s \xi_{B_s}(\theta_c)^\top] = \mathbb{E}\big[\delta_s\, \mathbb{E}[\xi_{B_s}(\theta_c)^\top \mid \delta_s, \theta_c]\big] = 0, \qquad \mathbb{E}[\xi_{B_s}(\theta_c)\delta_s^\top] = 0.$$

Therefore, computing $C_{s+1} = \mathbb{E}[\delta_{s+1}\delta_{s+1}^\top]$ from Eq. (14) gives

$$\begin{aligned} C_{s+1} &= \mathbb{E}[\delta_{s+1}\delta_{s+1}^\top] \\ &= \mathbb{E}\big[\big(A_c\delta_s - \eta P_{\mathrm{dom}}(\theta_c)\xi_{B_s}(\theta_c)\big)\big(A_c\delta_s - \eta P_{\mathrm{dom}}(\theta_c)\xi_{B_s}(\theta_c)\big)^\top\big] \\ &= A_c C_s A_c^\top + \eta^2 P_{\mathrm{dom}}(\theta_c)\Sigma_\xi(\theta_c)P_{\mathrm{dom}}(\theta_c). \end{aligned}$$

Thus, by the definition in Eq. (16),

$$C_{s+1} = A_c C_s A_c^\top + \eta^2 \Sigma_{\mathrm{dom}}(\theta_c). \tag{17}$$

Eq. (17) is the covariance recursion showing how the dominant noise covariance $\Sigma_{\mathrm{dom}}(\theta_c)$ induces the local displacement covariance through the local recursion.

Stationary displacement covariance. We now derive the stationary displacement covariance created by the local recursion around the reference point θ_c.

Iterating Eq. (17) gives

$$C_s = A_c^s C_0 (A_c^\top)^s + \eta^2 \sum_{\ell=0}^{s-1} A_c^\ell \Sigma_{\mathrm{dom}}(\theta_c)(A_c^\top)^\ell. \tag{18}$$

By Assumption 4 in Appendix B.2, $\rho(A_c) < 1$, so $A_c^s C_0 (A_c^\top)^s \to 0$, and the second term in Eq. (18) also converges as $s \to \infty$. We denote its limit by

$$C_{\mathrm{mb}}(\theta_c) := \lim_{s\to\infty} C_s$$

and call it the mini-batch-induced displacement covariance. Then $C_{\mathrm{mb}}(\theta_c)$ satisfies

$$C_{\mathrm{mb}}(\theta_c) = A_c C_{\mathrm{mb}}(\theta_c) A_c^\top + \eta^2 \Sigma_{\mathrm{dom}}(\theta_c). \tag{19}$$

Eq. (19) is a discrete Lyapunov equation for $C_{\mathrm{mb}}(\theta_c)$. Equivalently, $C_{\mathrm{mb}}(\theta_c)$ is given by the $s \to \infty$ limit of Eq. (18), namely

$$C_{\mathrm{mb}}(\theta_c) = \eta^2 \sum_{\ell=0}^{\infty} A_c^\ell \Sigma_{\mathrm{dom}}(\theta_c)(A_c^\top)^\ell. \tag{20}$$

Eq. (20) shows that $C_{\mathrm{mb}}(\theta_c)$ is formed by accumulating the dominant noise covariance $\Sigma_{\mathrm{dom}}(\theta_c)$ through the local recursion, with its value determined by the noise covariance, the local Hessian structure, and the learning rate. This covariance summarizes the effect of mini-batch noise in the dominant subspace.

Mini-batch-induced sharpness correction. Now set $C = C_{\mathrm{mb}}(\theta_c)$ in Eq. (5). Then the averaged gradient is

$$\mathbb{E}_\delta[\nabla L(\theta_c + \delta)] = \nabla L(\theta_c) + \frac{1}{2}\,\nabla_\vartheta \operatorname{Tr}(H(\vartheta)C_{\mathrm{mb}}(\theta_c))|_{\vartheta=\theta_c} + \mathcal{R}_3(\theta_c). \tag{21}$$

Eq. (21) shows that the mini-batch-induced displacement covariance $C_{\mathrm{mb}}(\theta_c)$ leaves an additional term in the averaged gradient. We refer to this term as the mini-batch-induced sharpness correction term. For the isotropic covariance in Section C.2, this term is proportional to $\nabla S_k(\theta_c)$. In contrast, $C_{\mathrm{mb}}(\theta_c)$ is generally not isotropic, so the correction term in Eq. (21) is not simply proportional to $\nabla S_k(\theta_c)$.

Therefore, in Section 5, we apply the corresponding sharpness correction to full-batch GD and compare the resulting corrected GD with mini-batch SGD. Specifically, Figure 4 compares full-batch GD, mini-batch SGD, and corrected GD, all starting from the same initialization, and shows that this correction reduces the sharpness gap between GD and SGD.

Appendix D. Experimental Details

In this section, we summarize the codebase, datasets, architectures, optimization settings, Hessian computation, and correction computation used in the main experiments.

D.1. Implementation

The experimental code is based on the supplementary implementation provided with Song et al. [29]. This implementation follows the experimental setup and codebase of Cohen et al. [3]. The experiments were run on an internal server with 8 NVIDIA RTX 3090 GPUs.

On top of this codebase, we added the estimation of the displacement covariance C_{mb}, the correction term b_{corr}, and corrected GD used in this paper. For Hessian eigenspace computation and higher-order automatic differentiation, we followed the numerical conventions of Cohen et al. [4]. In particular, for the Transformer experiments, we use a vanilla PyTorch LayerNorm implementation instead of the default PyTorch `nn.LayerNorm`, which was needed to compute the third-order derivatives in the correction term.

D.2. Datasets and Architectures

Table 1 summarizes the datasets and architectures used in our experiments.

Table 1: Dataset–architecture pairs used in our experiments.

Dataset	Task	Architecture
MNIST-5k [18]	10-class	3-layer MLP with two width-200 hidden layers and tanh activation
CIFAR10-5k [16]	10-class	3-layer CNN with width 32, ReLU activation, max pooling, and a linear classifier head
SST2-1k [28]	Binary	2-layer Transformer encoder with hidden dimension 64, 8 attention heads, ReLU activation, mean pooling, and a linear classifier head

For MNIST-5k and CIFAR10-5k, we use the first 5,000 training examples. For SST2-1k, we use the first 1,000 training examples. All experiments use only the training split. The MLP and CNN settings follow Cohen et al. [3], while the Transformer setting follows Damian et al. [5].

D.3. Training Setup

All main runs use vanilla mini-batch SGD or full-batch GD. The learning rate is kept constant, and we do not use momentum, weight decay, warmup, learning-rate decay, or early stopping. The default loss is the MSE loss with one-hot labels. Gradients, Hessian-vector products, Hessian eigenvalues, and covariance estimates are all computed using the $1/N$-normalized training loss.

Unless stated otherwise, we follow the stable learning-rate (GF) regime of Song et al. [29]. We use a batch size of 50, with learning rate 0.01 for the MLP and 0.001 for the CNN and Transformer. Each main run is trained for 20,000 steps.

D.4. Hessian and Correction Computation

Hessian-related quantities are computed at analysis checkpoints placed every 100 training steps. At each checkpoint, we compute the top Hessian eigenpairs using Hessian-vector products and LOBPCG [15]. We compute 20 eigenpairs at each checkpoint and use the top-k eigenvectors as the dominant-subspace basis. We use $k = 10$ for MNIST-5k and CIFAR10-5k, and $k = 2$ for SST2-1k.

At the same checkpoint, we estimate the mini-batch noise covariance on the dominant Hessian basis using 100 newly sampled mini-batches of size 50. The quantities C_{mb} and b_{corr} are then computed as described in Appendix C.3. In corrected GD, b_{corr} is recomputed at each checkpoint and kept fixed until the next checkpoint.

D.5. Reported Quantities and Baselines

At the analysis checkpoints above, we record training loss, gradients, top Hessian eigenvalues, top-k sharpness S_k, and dominant alignment χ_k. The main comparisons are mini-batch SGD, full-batch GD, and corrected GD. In the projected-update experiments, we also report Dom-SGD and Bulk-SGD. In the perturbation experiments, we compare dominant perturbation with random perturbation matched in rank and total variance.

Appendix E. Additional Experiments

In this section, we provide additional experimental results complementing Section 5. We first examine how the sharpness evolution changes when the batch size is changed. We then reverse the direction of the correction term b_{corr} to check whether the sharpness-reducing effect depends on the direction of the correction.

E.1. Batch Size Sweep

We compare mini-batch SGD, full-batch GD, and corrected GD obtained by applying the sharpness correction term to full-batch GD under different batch sizes in Figure 5.

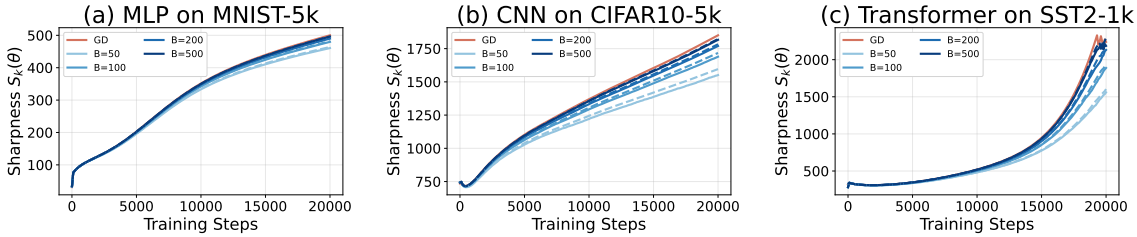

Figure 5: **Corrected GD across batch sizes.** (a) MLP on MNIST-5k, (b) CNN on CIFAR10-5k, and (c) Transformer on SST2-1k. Solid lines denote mini-batch SGD, and dashed lines denote corrected GD. As the batch size increases, the S_k curve of SGD moves closer to full-batch GD, and corrected GD follows the same trend.

E.2. Correction Direction

We reverse the direction of the correction term b_{corr} and compare it with the original corrected GD. This experiment shows that sharpness reduction depends on the direction of the derived correction term. Figures 6 and 7 show the corresponding sharpness and loss curves.

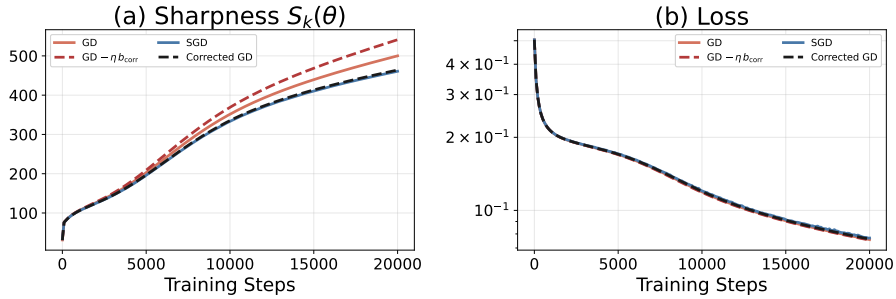

Figure 6: **Effect of the correction direction on MLP / MNIST-5k.** We compare GD, SGD, corrected GD, and GD with the reversed correction. (a) Top-k sharpness S_k. (b) Training loss. The derived correction lowers sharpness relative to GD, whereas reversing its direction increases sharpness relative to GD. The loss curves remain largely unchanged.

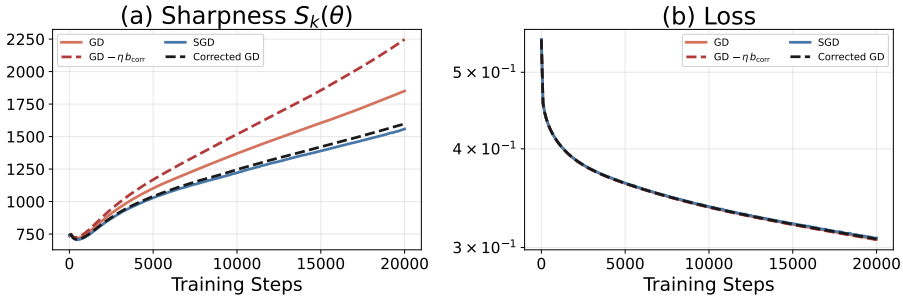

Figure 7: **Effect of the correction direction on CNN / CIFAR10-5k.** We compare GD, SGD, corrected GD, and GD with the reversed correction. (a) Top-k sharpness S_k. (b) Training loss. The derived correction lowers sharpness relative to GD, whereas reversing its direction increases sharpness relative to GD. The loss curves remain largely unchanged.