Unveiling the Hessian's Connection to the Decision Boundary

Mahalakshmi Sabanayagam*
School of Computation, Information and Technology, Technical University of Munich, Germany
sabanaya@cit.tum.de

Freya Behrens*
Statistical Physics of Computation Lab, École Polytechnique Fédérale de Lausanne, Switzerland
freya.behrens@epfl.ch

Urte Adomaityte
Department of Mathematics, King's College London, United Kingdom
urte.adomaityte@kcl.ac.uk

Anna Dawid†
Center for Computational Quantum Physics, Flatiron Institute, USA
adawid@flatironinstitute.org

Abstract

Understanding the properties of well-generalizing minima is at the heart of deep learning research. On the one hand, the generalization of neural networks has been connected to the decision boundary complexity, which is hard to study in the high-dimensional input space. Conversely, the flatness of a minimum has become a controversial proxy for generalization. In this work, we provide the missing link between the two approaches and show that the Hessian top eigenvectors characterize the decision boundary learned by the neural network. Notably, the number of outliers in the Hessian spectrum is proportional to the complexity of the decision boundary. Based on this finding, we provide a new and straightforward approach to studying the complexity of a high-dimensional decision boundary.

1 Introduction

The loss landscape of a deep neural network is a high-dimensional non-convex object exhibiting multiple equivalent local minima and saddle points [5, 8, 7, 37, 2]. The complex geometry of the loss landscape makes it notoriously difficult to analyze. In the context of gradient descent-based optimization, it is widely observed that the network converges to a local minimum that generalizes reasonably well [23]. Still, the properties of well generalizing minima are highly debated.

To understand those properties, some works study the decision boundary corresponding to a given minimum. Researchers often follow Occam's razor by assuming that among minima with similarly high training accuracy, the ones with simpler decision boundaries will have a higher test accuracy [17]. Other works study minima by analyzing their curvature in the model parameter space and developing heuristics indicating their generalization abilities. A few notable results suggest that flat minima

*Equal contribution.
†Corresponding author.

Mathematics of Modern Machine Learning Workshop at NeurIPS 2023.

generalize better than sharp minima [23, 42, 21, 19]. One approach to analyzing the curvature of the minimum is through the Hessian of the training loss. In particular, a sum of Hessian eigenvalues (trace) is sometimes used as a straightforward metric for generalization [20, 23]. However, these results are extensively contested and discussed [38, 22, 44, 35, 3], since flatness is not a well-defined concept in non-convex landscapes of deep models [10].

Despite the intuitive understanding that the simple decision boundary and a properly defined flatness of minima together promote good generalization of neural networks, to the best of our knowledge, no explicit connection between the two has been established so far. Advancing an understanding of this connection is precisely the goal of this work. To do so, we take a closer look at properties of the Hessian that are observed to be universal across different deep learning setups. Firstly, the spectrum of the Hessian at a minimum separates into the bulk centered around zero and a few outliers, whose number is roughly equal to the number of classes in the data [37, 38, 15, 31, 32]. We ask *what is the significance of those outliers, and why is their number approximately equal to the number of classes?* Secondly, *why does the gradient information reside in a small subspace spanned by the Hessian top eigenvectors* as noted by Gur-Ari et al. [18]?

In our work, we give an understanding of these properties by revealing their connection to the decision boundary. In particular, we compare the gradient directions of the loss for individual data points with the Hessian eigenvectors and see that they align when the samples are at the decision boundary. As a consequence, we propose a new generalization measure and a margin estimation technique that show promising empirical success in capturing the generalization of neural networks in Appendix I.

Contributions. We perform a rigorous numerical analysis of the deep neural network loss landscape for classification tasks using the Hessian of training loss, and we observe the following:
(1) The top eigenvectors of the Hessian of training loss encode the decision boundary learned by the neural network. In particular, there is a clear information separation across the eigenvectors, which encode separate sections of the decision boundary.
(2) The number of encoding eigenvectors is usually equal to the number of spectrum outliers, which is directly proportional to the complexity of the decision boundary. To elaborate, more eigenvectors are needed to encode a complex, highly non-linear decision boundary than a simpler counterpart.

2 Definitions and numerical details

We consider a C class classification problem with training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \dots, C\}$ is the class label. Let the neural network be $f_\theta : \mathbb{R}^d \to \mathbb{R}^C$ parameterized by $\theta \in \mathbb{R}^p$ where we focus on the over-parameterized setting, that is, $p \gg nd$. We obtain the class prediction as $\hat{y}_i = \arg\max f_\theta(x_i)$. We train the network f_θ using stochastic gradient descent (SGD) and cross-entropy loss $\mathcal{L}(\theta; \mathcal{D})$. We define the geometric complexity of the decision boundary as the number of its linear segments following Kienitz et al. [24].

At the heart of our work are the *gradients of loss of individual data points* $g_\theta : \mathbb{R}^d \to \mathbb{R}^p$ of x, where for labels we take the current prediction of f_θ at an input x:

$$g_\theta(x) = \frac{\partial}{\partial \theta} \mathcal{L}\left(\theta; \{x, \hat{y}\}\right) . \tag{1}$$

Note that x can be taken outside the training set. Finally, let the Hessian matrix of the training loss on data $\mathcal{D}$ be the square matrix $H \in \mathbb{R}^{p \times p}$ such that λ_i and $v_i \in \mathbb{R}^p$ denote the eigenpair of H. Therefore, the top k Hessian eigenvectors correspond to the first k largest Hessian eigenvalues. Then, we define the *alignment* between the gradient $g_\theta(x)$ of an input x with the eigenvector v_i in terms of cosine similarity as

$$\mathcal{A}_i(x) = \frac{\langle g_\theta(x), v_i \rangle}{\|g_\theta(x)\| \|v_i\|}, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ is the scalar product, and $\| \cdot \|$ is the Euclidean norm of the vector. Note that this alignment crucially depends on θ in the parameter space from which the Hessian H is derived.

Datasets and architectures. We consider five various two-dimensional (2D) simulated datasets and validate our findings on real datasets such as *Iris*, *MNIST*, and *CIFAR-10*. We study both two-layered fully-connected neural networks and convolutional neural networks. Detailed descriptions of the

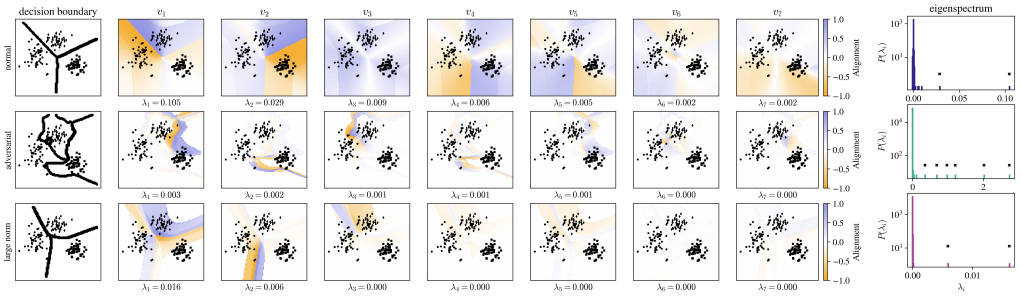

Figure 1: **Decision boundaries of different complexities for *gaussian*.** Alignment plots and histograms of the Hessian spectra for models obtained from normal training, an adversarial initialization [27], and a large norm initialization.

datasets and the models are provided in the code available here. In the main body of the manuscript, we focus on *gaussian* with three classes sampled from Gaussian mixtures. We present results for the toy example and other setups in Appendices A-B and K. When the models and datasets are of tractable sizes, we compute the Hessian exactly using the `torch.autograd` module from PyTorch [33], otherwise we approximate Hessian based on Golmant et al. [16].

We measure the alignment of $g_\theta(x)$ and each Hessian eigenvector for a converged network $f_{\hat{\theta}}$ and conduct an extensive empirical analysis that leads to the following results.

3 Results

Top Hessian eigenvectors encode the decision boundary. We plot the alignment of gradients $g_\theta(x)$ and each of the top $k = 7$ eigenvectors for the 2D *gaussian* dataset for three different decision boundary in Figure 1. For the topmost eigenvectors, *we observe a close-to-one absolute alignment with gradients of loss of the points on the decision boundary learned by the network.* Moreover, for these points, we see a transition from maximal positive to negative cosine similarity values, i.e., a switch of the alignment sign. These results hold for all other 2D datasets as presented in Appendix B.

We strengthen this observation mathematically by expanding the loss around a minimum θ^* using the second-order Taylor approximation at $\theta^* + \Delta\theta$ and considering $\Delta\theta = \frac{g_\theta(x)}{||g_\theta(x)||}$, resulting in

$$\mathcal{L}\left(f\left(\theta^*, \mathcal{X}\right) + \nabla_\theta f\left(\theta^*, \mathcal{X}\right)^T \frac{g_\theta(x)}{||g_\theta(x)||}, \mathcal{Y}\right) = \frac{1}{2}\sum_{i=1}^{p} \lambda_i \mathcal{A}_i(x)^2. \tag{3}$$

Let's consider the maxima of both sides of (3) with respect to x. On the left side, the maximum loss is when the gradient of x aligns with the direction of the steepest ascent of $f(\theta^*, \mathcal{X})$ that happens for x at the decision boundary. On the right side, the maximal value of the sum occurs for $g_\theta(x)$ perfectly aligned with the Hessian top eigenvector with the largest eigenvalue λ_1. From this, we infer that the alignment of x with the top Hessian eigenvectors is larger for x near the boundary than for data points farther away, explaining our numerical observation. A detailed analysis is provided in Appendix C.

Additionally, we see that *each top eigenvector may capture only a section of the complete boundary.* The information on the sections of decision boundary can be well separated across eigenvectors (see Figure 1). Moreover, the alignment does not necessarily switch between extreme values $+1$ and -1 across the decision boundary. *The exact values do not seem informative in contrast to the sign switch itself.* Finally, the largest alignment across the input space is always for points on the boundary.

We perform a similar step-by-step analysis of the alignment using a toy example in Appendix A. We show that the cosine similarity is more informative than the scalar product in Appendix D. Moreover, in Appendix E, we show that *the first few eigenvectors are sufficient to encode the entire decision boundary*, and the other directions in the parameter space do not exhibit the same property. Interestingly, when we analyze the Hessian with respect to the loss that only considers a specific class c, we observe that the top eigenvectors are now restricted to the boundaries that are relevant to deciding the "all-against-one" for the class c (Appendix F). In Appendix G, we show that this observation is invariant to the architecture, loss function and optimizer. Finally, we highlight that it holds throughout the training (Appendix H).

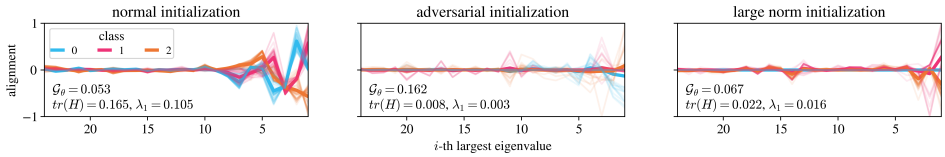

Figure 2: **Alignment of all training data with the top** 25 **Hessian eigenvectors for *gaussian* with classes** $\{0, 1, 2\}$ **and different initializations.** The dark lines show the mean of each class alignment.

A complex boundary is characterized by many eigenvectors. Past works indicate that the number of outliers in the Hessian spectrum is roughly equal to the number of classes in the dataset. However, we hypothesize that *the number of outliers depends on the simplicity of the learned decision boundary.* An increased number of eigenvectors, corresponding to an increased number of outliers, is needed to characterize a more complex decision boundary.

To verify our hypothesis, we follow different training procedures to reach poorly generalizing minima which, by Occam's razor, usually imply complex decision boundaries. We use two such methods. One is an adversarial initialization as introduced by Liu et al. [27]. Briefly, the procedure consists in initializing the network with parameters θ that fit the data with random labels. We notice that such an initialization always exhibits a large L_2 norm. Therefore, another method we use consists in simply large norm initialization of the model. Usually, the adversarial and large norm initializations lead to much more complex and slightly more complex decision boundary than the regular initialization, respectively, as presented in the first column of Figure 1. We compute the alignment of gradients with the top Hessian eigenvectors corresponding to the outliers for all the initialization methods on *gaussian* as shown in Figure 1, which demonstrates *more eigenvectors are needed to describe the learned decision boundary from both adversarial and large norm initializations.*

To complete the picture, the last column of Figure 1 shows the histogram of Hessian eigenvalues for *gaussian* dataset illustrating that *different training procedures lead to a different number of outliers*. In particular, normal training leads to 2 outliers following the conjectures from the past works, whereas the adversarial initialization shows more outliers. It is important to note that the number of the top Hessian eigenvectors that encode sections of the decision boundary does not correspond one-to-one to the number of outliers in the spectra.

Localization of gradients happens for well generalizing minima. We plot the alignment of gradients of loss of training samples with the top Hessian eigenvectors in Figure 2 and confirm that our main observation holds also for this subset of the input space: indeed, a more complex decision boundary leads to a larger number of Hessian eigenvectors with non-zero alignment with training gradients. Moreover, *for simpler decision boundary aka well generalizing minimum (normal initialization) the gradients are much more localized in the Hessian space, that is, the alignment is significantly greater than* 0 *only for the top eigenvectors, than in the other initializations.* Interestingly, we also see that gradients at the well generalizing minimum are more aligned with one another according to their classes.

Validation of our results on real data We validate our observations on high dimensional datasets and more complex architecture trained with both normal and adversarial initialization. We obtain the results for *Iris*, *MNIST* and *CIFAR-10*. We present the results for *MNIST-017*, where numbers indicate selected classes of digits, and provide the full analysis in Appendix K. We visualize the dataset using t-SNE in Figure 3 and color code the alignment between the training gradients and top Hessian eigenvectors. While t-SNE does not necessarily have the same data representation as a trained neural network, the alignment behavior for models obtained with the normal and adversarial initialization training follows the one for the 2D datasets. It suggests that *the top eigenvectors encode the decision boundary*. Finally, we also see *a higher number of outliers in the eigenspectrum for complex boundary* (last column of Figure 3).

With this understanding, we propose a generalization measure that quantifies the complexity of the decision boundary and shows promising preliminary empirical results. To avoid overlooking the simplicity bias when it hurts generalization, we propose a novel margin estimation technique of the decision boundary using the alignment with the top Hessian eigenvector. We discuss these aspects elaborately with extensive experiments in Appendix I. While our results are promising, it is crucial to analytically understand the connection between the Hessian eigenvectors and the decision

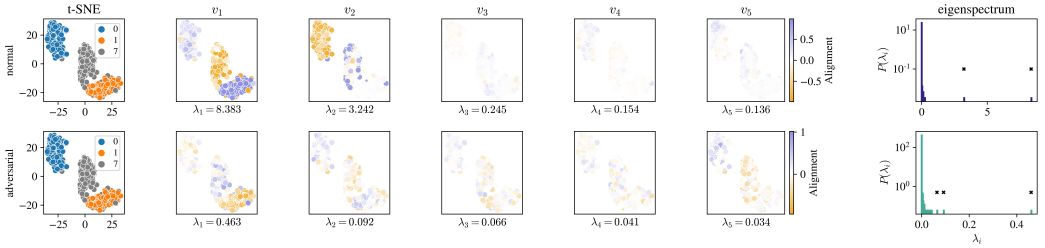

Figure 3: **Normal and adversarial initialization training for *MNIST-017* with t-SNE visualization.** It shows the alignments of gradients with the top 5 eigenvectors. *(Last column)* The eigenspectrum.

boundary, also with the Hessian eigenvalues if any. Nevertheless, our results inspire techniques to improve pruning, fight catastrophic forgetting by freezing the parameters corresponding to the decision boundary, assess the uncertainty of the prediction using the distance to the closest decision boundary, etc.

Acknowledgments and Disclosure of Funding

We thank Tony Bonnaire, Francesa Mignacco, Stefani Karp, Yatin Dandi, Artem Vysogorets, Boris Hanin, and Julia Kempe for useful discussions. This work started as an open problem posed by A.D. during the 2022 Les Houches Summer School on Statistical Physics and Machine Learning organized by Lenka Zdeborová and Florent Krzakala.

M.S. is supported by the German Research Foundation (Research Training Group GRK 2428). A.D. acknowledges the financial support from the Foundation for the Polish Science. The Flatiron Institute is a division of the Simons Foundation.

References

[1] Agarwal, N., Bullins, B., and Hazan, E. Second-order stochastic optimization for machine learning in linear time. *J. Mach. Learn. Res.*, 18:1–40, 2017. ISSN 15337928.

[2] Alain, G., Roux, N. L., and Manzagol, P.-A. Negative eigenvalues of the Hessian in deep neural networks, 2018. URL `https://openreview.net/forum?id=S1iiddyDG`.

[3] Andriushchenko, M., Croce, F., Müller, M., Hein, M., and Flammarion, N. A modern look at the relationship between sharpness and generalization, 2023. URL `https://arxiv.org/abs/2302.07011`.

[4] Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. A closer look at memorization in deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/arpit17a.html`.

[5] Auer, P., Herbster, M., and Warmuth, M. K. K. Exponentially many local minima for single neurons. In Touretzky, D., Mozer, M., and Hasselmo, M. (eds.), *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995. URL `https://proceedings.neurips.cc/paper/1995/file/3806734b256c27e41ec2c6bffa26d9e7-Paper.pdf`.

[6] Chatterjee, S. and Zielinski, P. On the generalization mystery in deep learning, 2022. URL `https://arxiv.org/abs/2203.10036`.

[7] Choromanska, A., Henaff, M., Mathieu, M., Ben Arous, G., and LeCun, Y. The loss surfaces of multilayer networks. In *AISTATS 2015 - 18th Int. Conf. Artif. Intell. Stat.*, volume 38, pp. 192–204. PMLR, 2015. URL `http://proceedings.mlr.press/v38/choromanska15.html`.

[8] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *NIPS 2014*

- *Adv. Neural. Inf. Process. Syst.*, volume 27, pp. 2933–2941, 2014. URL https://papers.nips.cc/paper/2014/hash/17e23e50bedc63b4095e3d8204ce063b-Abstract.html.

[9] Dawid, A., Huembeli, P., Tomza, M., Lewenstein, M., and Dauphin, A. Hessian-based toolbox for reliable and interpretable machine learning in physics. *Mach. Learn.: Sci. Technol.*, 3:015002, 2022. doi: 10.1088/2632-2153/ac338d. URL https://doi.org/10.1088/2632-2153/ac338d.

[10] Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1019–1028. PMLR, 06–11 Aug 2017.

[11] Dombrowski, A.-K., Alber, M., Anders, C., Ackermann, M., Müller, K.-R., and Kessel, P. Explanations can be manipulated and geometry is to blame. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/bb836c01cdc9120a9c984c525e4b1a4a-Paper.pdf.

[12] Dombrowski, A.-K., Anders, C. J., Müller, K.-R., and Kessel, P. Towards robust explanations for deep neural networks. *Pattern Recognition*, 121:108194, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.108194. URL https://www.sciencedirect.com/science/article/pii/S0031320321003769.

[13] Feng, Y. and Tu, Y. The activity-weight duality in feed forward neural networks: The geometric determinants of generalization, 2022.

[14] Fort, S. and Ganguli, S. Emergent properties of the local geometry of neural loss landscapes. *CoRR*, abs/1910.05929, 2019. URL http://arxiv.org/abs/1910.05929.

[15] Ghorbani, B., Krishnan, S., and Xiao, Y. An investigation into neural net optimization via Hessian eigenvalue density. In *ICML 2019 - 36th Int. Conf. Mach. Learn.*, volume 97, pp. 2232–2241. PMLR, 2019. ISBN 9781510886988. URL http://proceedings.mlr.press/v97/ghorbani19b.html.

[16] Golmant, N., Yao, Z., Gholami, A., Mahoney, M., and Gonzalez, J. Efficient PyTorch Hessian eigendecomposition, October 2018. URL https://github.com/noahgolmant/pytorch-hessian-eigenthings.

[17] Guan, S. and Loew, M. Analysis of generalizability of deep neural networks based on the complexity of decision boundary. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 101–106. IEEE, 2020.

[18] Gur-Ari, G., Roberts, D. A., and Dyer, E. Gradient descent happens in a tiny subspace, 2019. URL https://openreview.net/forum?id=ByeTHsAqtX.

[19] He, H., Huang, G., and Yuan, Y. Asymmetric valleys: Beyond sharp and flat local minima. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/01d8bae291b1e4724443375634ccfa0e-Paper.pdf.

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

[21] Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. In *UAI 2018 - 34th Conf. Uncertain. Artif. Intell.*, volume 2, pp. 876–885, 2018. ISBN 9781510871601. URL https://arxiv.org/abs/1803.05407v3.

[22] Jastrzebski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkgEaj05t7.

[23] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. 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.

[24] Kienitz, D., Komendantskaya, E., and Lones, M. Comparing complexities of decision boundaries for robust training: A universal approach. In Wang, L., Gall, J., Chin, T.-J., Sato, I., and Chellappa, R. (eds.), *Computer Vision – ACCV 2022*, pp. 627–645, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-26351-4.

[25] Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *ICML 2017 - 34th Int. Conf. Mach. Learn.*, volume 70, pp. 1885–1894. PMLR, 2017. URL http://proceedings.mlr.press/v70/koh17a.html.

[26] LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. In Touretzky, D. (ed.), *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

[27] Liu, S., Papailiopoulos, D., and Achlioptas, D. Bad global minima exist and SGD can reach them. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8543–8552. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/618491e20a9b686b79e158c293ab4f91-Paper.pdf.

[28] Madras, D., Atwood, J., and D'Amour, A. Detecting extrapolation with local ensembles. In *ICLR 2020 - Int. Conf. Learn. Represent.*, 2020. URL https://openreview.net/forum?id=BJl6bANtwH.

[29] Moosavi-Dezfooli, S. M., Fawzi, A., Uesato, J., and Frossard, P. Robustness via curvature regularization, and vice versa. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9070–9078, Los Alamitos, CA, USA, Jun 2019. IEEE Computer Society. doi: 10.1109/CVPR.2019.00929. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00929.

[30] Nakkiran, P., Kaplun, G., Kalimeris, D., Yang, T., Edelman, B. L., Zhang, F., and Barak, B. SGD on neural networks learns functions of increasing complexity. In *NeurIPS 2019 (spotlight)*, volume abs/1905.11604, 2019. URL http://arxiv.org/abs/1905.11604.

[31] Papyan, V. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5012–5021. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/papyan19a.html.

[32] Papyan, V. Traces of class/cross-class structure pervade deep learning spectra. *J. Mach. Learn. Res.*, 21(1), 2020. ISSN 1532-4435. doi: 10.5555/3455716.3455968.

[33] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[34] Pearlmutter, B. A. Fast exact multiplication by the hessian. *Neural Computation*, 6:147–160, 1994. doi: 10.1162/neco.1994.6.1.147.

[35] Petzka, H., Kamp, M., Adilova, L., Sminchisescu, C., and Boley, M. Relative flatness and generalization. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=sygvo7ctb_.

[36] Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., and Kohli, P. Adversarial robustness through local linearization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/0defd533d51ed0a10c5c9dbf93ee78a5-Paper.pdf.

[37] Sagun, L., Bottou, L., and LeCun, Y. Eigenvalues of the Hessian in deep learning: Singularity and beyond, 2017. URL https://openreview.net/forum?id=B186cP9gx.

[38] Sagun, L., Evci, U., Güney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the Hessian of over-parametrized neural networks. In *ICLR 2018 - 6th Int. Conf. Learn. Represent.*, 2018. URL https://openreview.net/forum?id=rJrTwxbCb.

[39] Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *CoRR*, abs/2006.07710, 2020. URL https://arxiv.org/abs/2006.07710.

[40] Srinivas, S., Matoba, K., Lakkaraju, H., and Fleuret, F. Efficiently training low-curvature neural networks, 2022. URL https://arxiv.org/abs/2206.07144.

[41] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017.

[42] Wu, L., Zu, Z., and E, W. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv:1706.10239*, 2017. URL https://arxiv.org/abs/1706.10239.

[43] Yu, S., Yao, Z., Gholami, A., Dong, Z., Mahoney, M. W., and Keutzer, K. Hessian-aware pruning and optimal neural implant. *CoRR*, abs/2101.08940, 2021. URL https://arxiv.org/abs/2101.08940.

[44] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, Feb 2021. ISSN 0001-0782. doi: 10.1145/3446776. URL https://doi.org/10.1145/3446776.

[45] Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. E., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhang19p.html.

Supplementary material

accompanying "*Unveiling the Hessian's Connection to the Decision Boundary*"

We provide the following results in the supplementary material.

- **Section A**: Illustration of our Hessian-gradient analysis on a toy example
- **Section B**: The top Hessian eigenvectors and decision boundary for the additional two-dimensional datasets
- **Section C**: Theoretical analysis
- **Section D**: Alignment of vectors: cosine similarity vs scalar product
- **Section E**: Directions other than the top Hessian eigenvectors do not align with the decision boundary
- **Section F**: Decision boundary per class
- **Section G**: Results are invariant to an architecture, loss, and optimizer: *gaussian*
- **Section H**: Decision boundaries during the training
- **Section I**: Generalization measure and margin estimation technique for simulated datasets
- **Section J**: Generalization measure is invariant to model reparameterization
- **Section K**: Validation of the observations on real datasets (*Iris*, *MNIST*, and *CIFAR-10*)
- **Section L**: Discussion and related works
- **Section M**: The gradient covariance matrix vs the Hessian at the minimum

A Our approach using a toy example

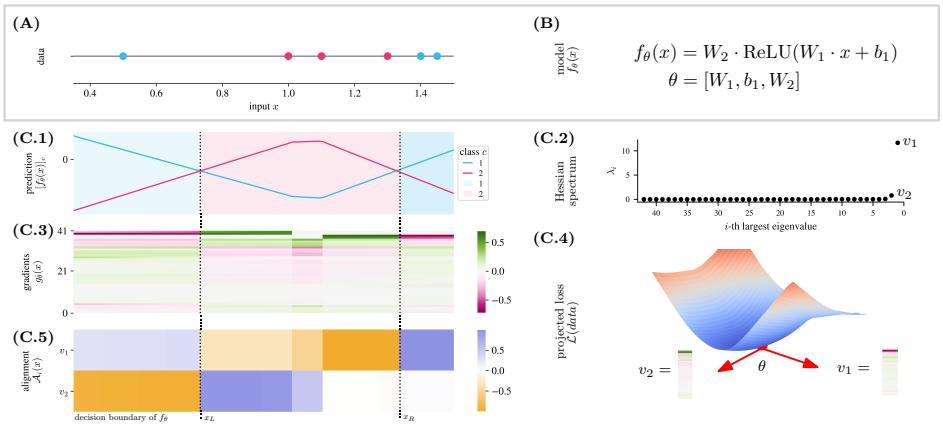

Figure 4: **Analysis of the Hessian. (A)** 1D toy dataset with 5 input points and 2 classes {pink, cyan}. **(B)** A model f_θ parameterized by θ that takes an input x and returns logit probabilities for each class. **(C.1)** Predictions of $f_{\hat{\theta}}$ across the input space with $\hat{\theta}$ being a specific set of parameters that correctly classify the training data. **(C.2)** There are two outliers in the Hessian eigenspectrum of the training loss calculated at the minimum $\hat{\theta}$. They correspond to the eigenvectors v_1 and v_2 that are directions in the parameter space shown in **(C.4)**. When measuring their cosine similarity with gradients of the loss of individual points from the input space **(C.3)**, we obtain the alignment in **(C.5)**. Each outlier encodes one section of the decision boundary of $f_{\hat{\theta}}$ respect to the data that induced the loss landscape.

To detail our approach, consider a simple two-layer fully-connected ReLU network f_θ trained to classify one-dimensional (1D) training data into two classes as presented in Figure 4 (A)-(B). In this

1D input space, the network learns a decision boundary located at two points, x_L and x_R (Figure 4 (C.1)). Consider gradients of the loss function of individual data points in the input space as in Figure 4 (C.3). Those gradients align with the directions in the parameter space corresponding to the largest increase of error on the data. Overall, the largest error is made on the data that is on the boundary by shifting it across the boundary. Moreover, gradients on the opposite sides of the boundary point in opposite directions as moving the boundary benefits samples from one class but hurts samples from another. Indeed, we see that gradients on either side of x_L and x_R point in opposite directions. These directions align with the top two Hessian eigenvectors v_1 and v_2 corresponding to the two outliers in its eigenspectrum (Figure 4 (C.2,C.4)). We see that gradients around x_R align perfectly with the top eigenvector v_1: The cosine similarity flips from -1 to 1 as the decision boundary is crossed (Figure 4 (C.5)). The gradients around x_L align with the second top eigenvector v_2. We conclude that the top Hessian eigenvectors encode separate pieces of the decision boundary learned by the network.

B The top Hessian eigenvectors and decision boundary for the additional simulated datasets

Within this work, we use five simulated 2D datasets: *gaussian* with three classes, concentric *circle* and *half-moon* datasets with two classes, *hierarchical gaussian* with four classes, and *checkerboard* dataset with two classes. Details on their generation can be found in the code available here (link is anonymized) (`src/datasets.py`).

In Figure 5, we present the alignment of the gradients of input samples with the top five Hessian eigenvectors for all the simulated datasets. We see consistently that *the top Hessian eigenvectors align maximally with the gradients of loss of samples at the boundary.*

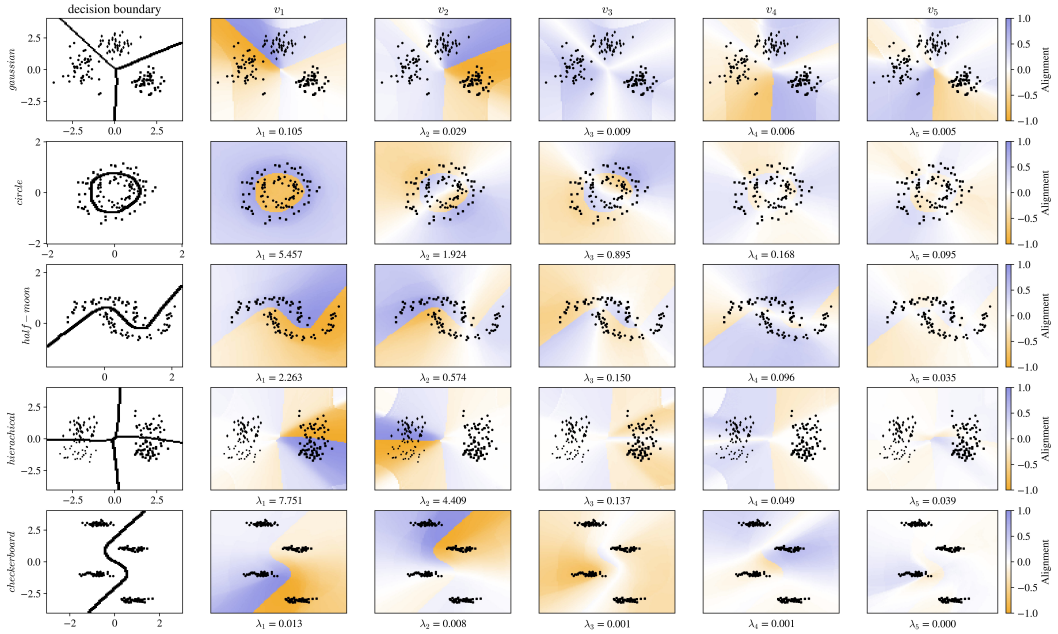

Figure 5: **Top Hessian eigenvectors and decision boundaries for all simulated datasets.** *(First column)* The decision boundary in the data space obtained by training a two-layered fully connected network. *(Other columns)* The alignment with the top five eigenvectors illustrates that the top eigenvectors encode the decision boundary.

C Theoretical analysis

We consider that the converged minimum $\theta := \theta^*$ is an exact minimum, meaning that the loss and its gradient at θ^* is zero, i.e., $\mathcal{L}(\theta^*; \mathcal{D}) = 0$ and $\nabla_\theta \mathcal{L}(\theta^*; \mathcal{D}) = \mathbf{0}$. Using this information, we expand the loss using the second-order Taylor's approximation at $\theta := \theta^*$.

$$\mathcal{L}(\theta^* + \Delta\theta; \mathcal{D}) = \mathcal{L}(\theta^*; \mathcal{D}) + \nabla_\theta \mathcal{L}(\theta^*; \mathcal{D})^T \Delta\theta + \frac{1}{2} \Delta\theta^T \nabla_\theta^2 \mathcal{L}(\theta^*; \mathcal{D}) \Delta\theta$$

$$\mathcal{L}(\theta^* + \Delta\theta; \mathcal{D}) = \frac{1}{2} \Delta\theta^T H_{\theta^*} \Delta\theta$$

H_{θ^*} is the Hessian of the training loss function evaluated at the minimum. We denote its eigenvectors and corresponding eigenvalues as v_i and λ_i. Now, let's consider $\Delta\theta := \frac{g_\theta(x)}{||g_\theta(x)||}$ to be a gradient of some input x in the dataset $\mathcal{D} := \{\mathcal{X}, \mathcal{Y}\}$, and an overparametrized classifier (e.g., neural network) with p parameters denoted by $f(\theta, \mathcal{X})$.

$$\begin{aligned}
\mathcal{L}\left(f\left(\theta^* + \frac{g_\theta(x)}{||g_\theta(x)||}, \mathcal{X}\right), \mathcal{Y}\right) &= \frac{1}{2} \frac{g_\theta(x)^T}{||g_\theta(x)||} \sum_{i=1}^{p} \lambda_i v_i v_i^T \frac{g_\theta(x)}{||g_\theta(x)||} \\
&= \frac{1}{2} \sum_{i=1}^{p} \lambda_i \frac{\langle g_\theta(x), v_i \rangle}{||g_\theta(x)||} \frac{\langle g_\theta(x), v_i \rangle}{||g_\theta(x)||} \\
\mathcal{L}\left(f\left(\theta^*, \mathcal{X}\right) + \nabla_\theta f\left(\theta^*, \mathcal{X}\right)^T \frac{g_\theta(x)}{||g_\theta(x)||}, \mathcal{Y}\right) &= \frac{1}{2} \sum_{i=1}^{p} \lambda_i \mathcal{A}_i(x)^2 \qquad (4)
\end{aligned}$$

From Equation (4), for the loss to have a maximal change, the gradient of x should be aligned with the direction of the steepest ascent of $f(\theta^*, \mathcal{X})$. This implies that moving data x in the direction of the gradient of $f(\theta^*, \mathcal{X})$ potentially changes the predicted class for x, thus increasing the loss. In other words, the alignment of the gradient of x with the function's gradient is high for x near the decision boundary. From this understanding, we infer that the alignment of x with the Hessian eigenvectors is larger for x near the boundary than data points farther away from the right-hand side of Equation (4). As the alignment $\mathcal{A}(x)$ considers a normalized $g_\theta(x)$, this variability comes only from a different alignment of $g_\theta(x)$ for x's close and far from the boundary with different Hessian eigenvectors. As stated multiple times, the behavior of the Hessian spectra in deep learning setups is universal. Its spectrum has a small number of positive non-zero eigenvalues, $\lambda_0, \lambda_1, \ldots, \lambda_t$, and the rest of the eigenvalues is close to zero. This, in turn, implies that the right side of the equation is large when $g_\theta(x)$ is aligned with the top Hessian eigenvectors with the largest eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_t$. This implication strengthens our numerical observations about the top Hessian eigenvectors being aligned with the gradients of loss of data at the boundary.

D Alignment of vectors: cosine similarity vs scalar product

In Equation (2), we have defined the *alignment* between the gradient $g_\theta(x)$ from Equation (1) of an input x with eigenvector v_i in terms of the cosine similarity as

$$\mathcal{A}_i(x) = \frac{\langle g_\theta(x), v_i \rangle}{\|g_\theta(x)\| \|v_i\|}, \qquad (5)$$

where $\langle \cdot, \cdot \rangle$ is the scalar product, and $\| \cdot \|$ is the Euclidean norm of the vector.

As the main findings of our work result from comparing gradients of loss of individual training samples with the Hessian top eigenvectors, one may ask why we chose cosine similarity as the measure of similarity between the vectors instead, e.g., of the scalar product itself, $\langle g_\theta(x), v_i \rangle$. We compare the two similarity metrics in Figure 6, where we immediately see the main weakness of the scalar product when it comes to studying the input space. First of all, due to the lack of gradients' normalization, the overlap highlights only points on the decision boundary. Large norm of gradients on the boundary dominates any existing alignment between the Hessian eigenvectors and gradients of samples far from the boundary. Secondly, contrary to the cosine similarity, the scalar product has no maximal value, which could guide the analysis of the decision boundary decomposition in terms of the Hessian eigenvectors.

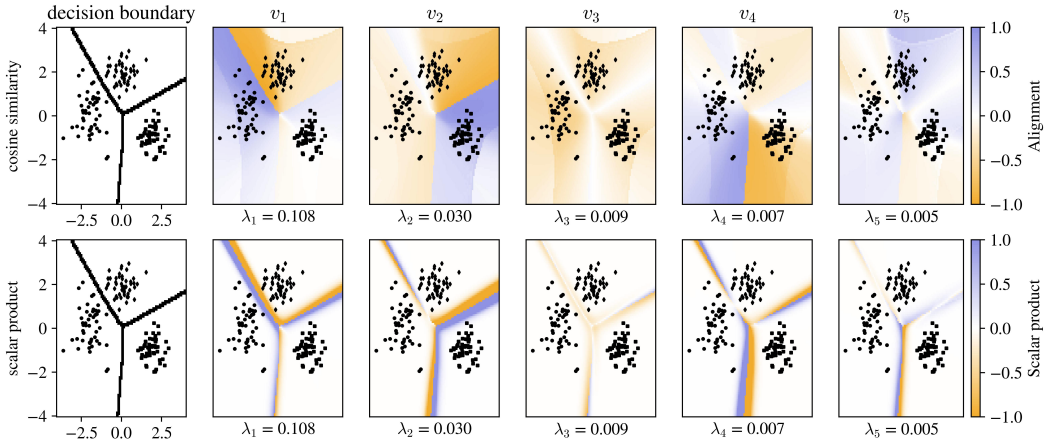

Figure 6: **Comparison of similarity metrics for *gaussian*.** *(Top)* Cosine similarity. *(Bottom)* Scalar product with a color bar limited to $[-1, +1]$ (without normalizing the scalar product values).

E Directions other than the top Hessian eigenvectors do not align with the decision boundary

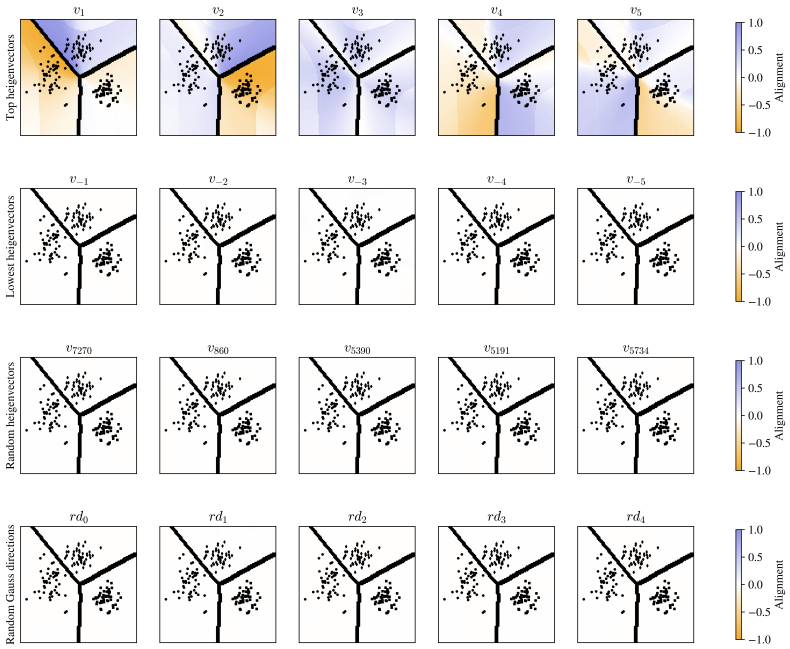

Figure 7: **Comparison of different directions in parameter space and their alignment with the gradients of input points in the 2D plane.** Alignment color normalized to the interval between $[-1, +1]$. We show the top eigenvectors $v_1, \ldots, v_5$ and compare them to the eigenvectors with the smallest eigenvalues where v_{-1} has the smallest eigenvalue and v_{-5} the fifth-smallest eigenvalue. We also sample some random directions from the Hessian eigenspace and finally compare to random directions in parameter space where each entry of the vector is sampled from a standard Gaussian.

To test our finding connecting the top Hessian eigenvectors and decision boundary, we check the alignment of gradients of individual input samples with vectors pointing in other directions in parameter space. In Figure 7 we compare the gradients' alignment with *(1)* the top five Hessian eigenvectors, *(2)* the bottom five Hessian eigenvectors (that correspond to the five smallest eigenvalues,

i.e., five largest negative eigenvalues), *(3)* five randomly selected Hessian eigenvectors, and *(4)* five random directions in parameter space where each vector element is sampled from a standard Gaussian. Within this comparison, the color bar is normalized across the plots to $[-1, +1]$. We immediately see that *directions other than the top Hessian eigenvectors are not aligned with gradients in the slightest.*

Interestingly, when we use separate color bars for each plot and repeat the comparison in Figure 8, we see that the random directions may sometimes reflect the alignment of the gradients resulting from the decision boundary. Tiny alignment values around 10^{-2} show, however, that it reflects the coherence of gradients rather than encodes relevant directions in parameter space. *The average of the maximal alignment of training gradients with five randomly sampled directions in the parameter space serves therefore as a threshold ϵ in Equation 6 for counting Hessian eigenvectors with "non-zero" alignment with gradients.*

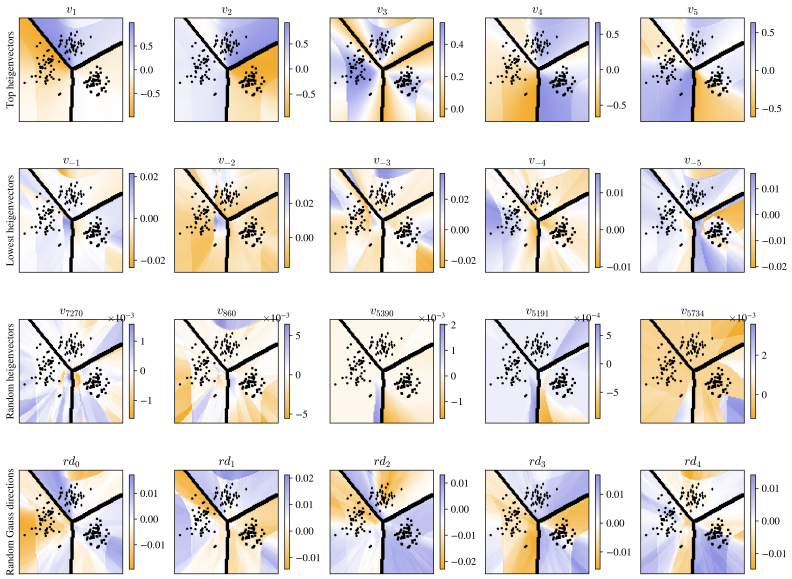

Figure 8: **Comparison of different directions in parameter space and their alignment with the gradients of input points in the 2D plane - The color bars are normalized per plot.** We show the top eigenvectors $v_1, \ldots, v_5$ and compare them to the eigenvectors with the smallest eigenvalues where v_{-1} has the smallest eigenvalue and v_{-5} the fifth-smallest eigenvalue. We also sample some random directions from the Hessian eigenspace and finally compare to random directions in parameter space where each entry of the vector is sampled from a standard Gaussian.

F Decision boundary per class

The Hessian eigenvectors crucially depend on the loss landscape, which in turn depends on the data. When we analyze the Hessian with respect to the loss that only considers a specific class c, we observe that *the top eigenvectors are now restricted to the boundaries that are relevant to deciding the "all-against-one" for the selected class c.* The results are presented in Figure 9.

G Results are invariant to an architecture, loss, and optimizer: *gaussian*

Here, we show that while the decision boundary may shift and the alignment values change, the connection between the topmost eigenvectors and *the decision boundary is invariant to the architecture* (Figure 10), *the choice of the optimizer* (Figure 11), and *loss function* (Figure 12).

Interestingly, while SGD, Adam, AdamW, and RMSprop in Figure 11 reach similar decision boundaries, values of alignment of gradients across the input space with the top eigenvectors vary significantly. In particular, in SGD, the non-zero alignment with the top eigenvectors is preserved for gradients far from the boundary. In Adam, AdamW, and RMSprop, the alignment goes quickly to

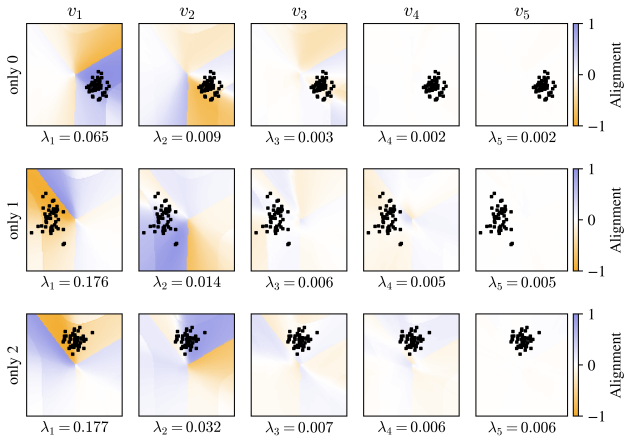

Figure 9: **Loss decomposed into separate classes.** The decision boundaries are equivalent to those from the top row of Figure 1. Here, the training loss is decomposed into the losses of individual training points associated with a given class. The Hessian eigenvectors look different for each decomposition, and the top eigenvectors exactly show the decision boundary enclosing this class. Only the relevant class is shown.

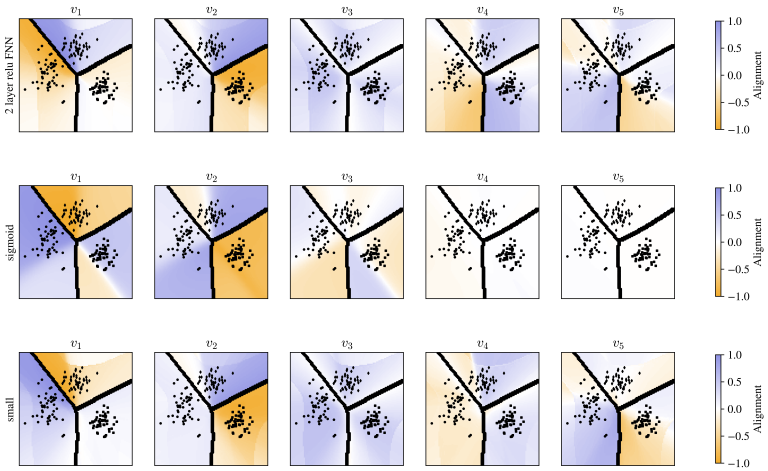

Figure 10: **Top eigenvectors of the Hessian for alternative model architectures.** *(First row)* A two-layer neural network with 100 neurons per layer and the ReLU activation function. *(Second row)* The same with sigmoid activations. *(Third row)* The same as first but with 50 neurons per layer instead.

zero with the distance from the boundary. It may suggest that in such cases, the gradient-based interpretation methods such as influence functions [25] may fail to find examples that are similar to a sample far from the boundary. At the same time, the connection between the decision boundary itself and the top Hessian eigenvectors is preserved regardless of the optimizer.

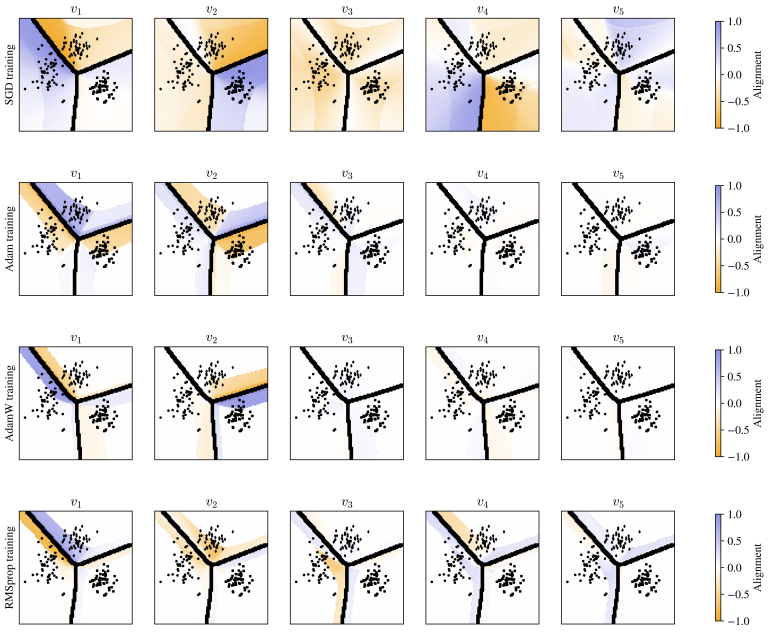

Figure 11: **Top eigenvectors of the Hessian for different optimizers:** *(First row)* SGD, *(Second row)* Adam, *(Third row)* AdamW, and *(Fourth row)* RMSprop with the same learning rate of 0.2 and a batch size of 64. For optimizers besides SGD, the boundaries are more "clear cut"; The gradient at many places in the input space has zero alignment with their counterparts on the boundary.

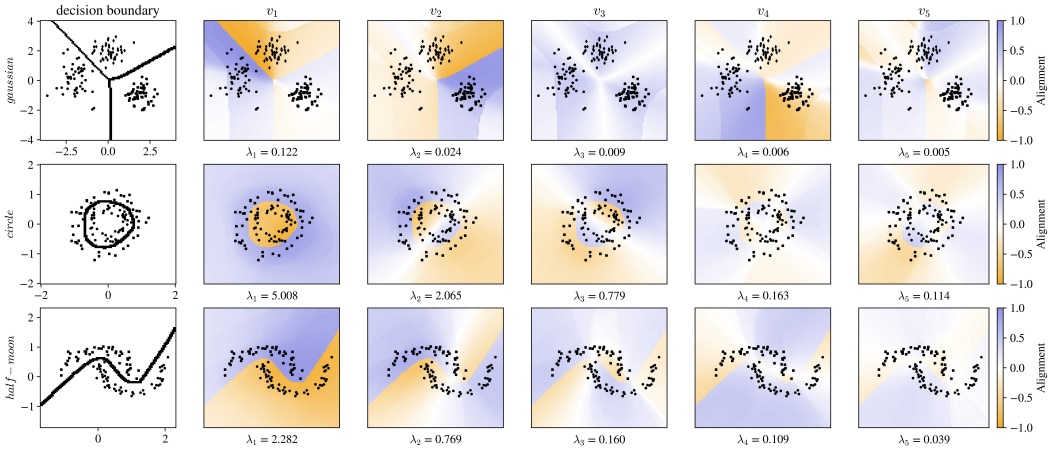

Figure 12: **Top eigenvectors of the Hessian for the negative log-likelihood loss (NLLLoss):** *(Top) gaussian* dataset. *(Middle) circle* dataset. *(Bottom) half-moon* dataset.

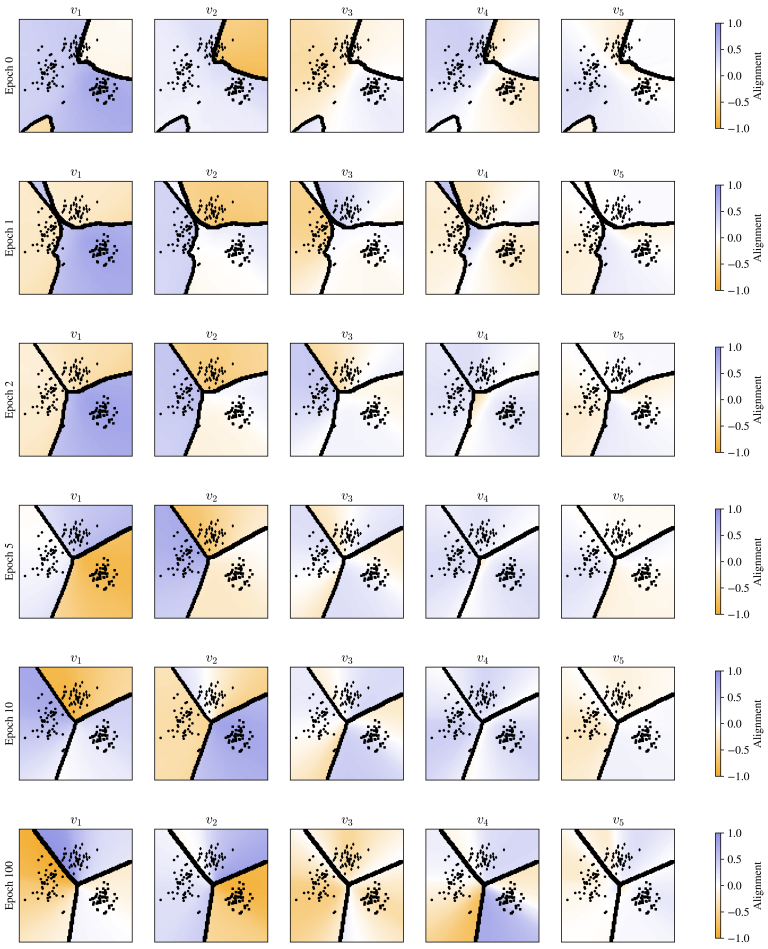

Figure 13: **Top Hessian eigenvectors encode boundaries also away from the minimum.** The overlap plots during different epochs in for normal training on *gaussian*. Epoch 0 is the boundary at initialization before training.

H Decision boundaries during the training

Interestingly, we see that *the top Hessian eigenvectors encode the decision boundary also away from the minimum during the training dynamics.* We believe that the gradient covariance matrix will not exhibit this behavior since it is not at the minimum. We present the usual alignment analysis between gradients of loss of input samples and the top five Hessian eigenvectors for selected epochs of the regular training in Figure 13 and training starting from the adversarial initialization of Liu et al. [27] in Figure 14.

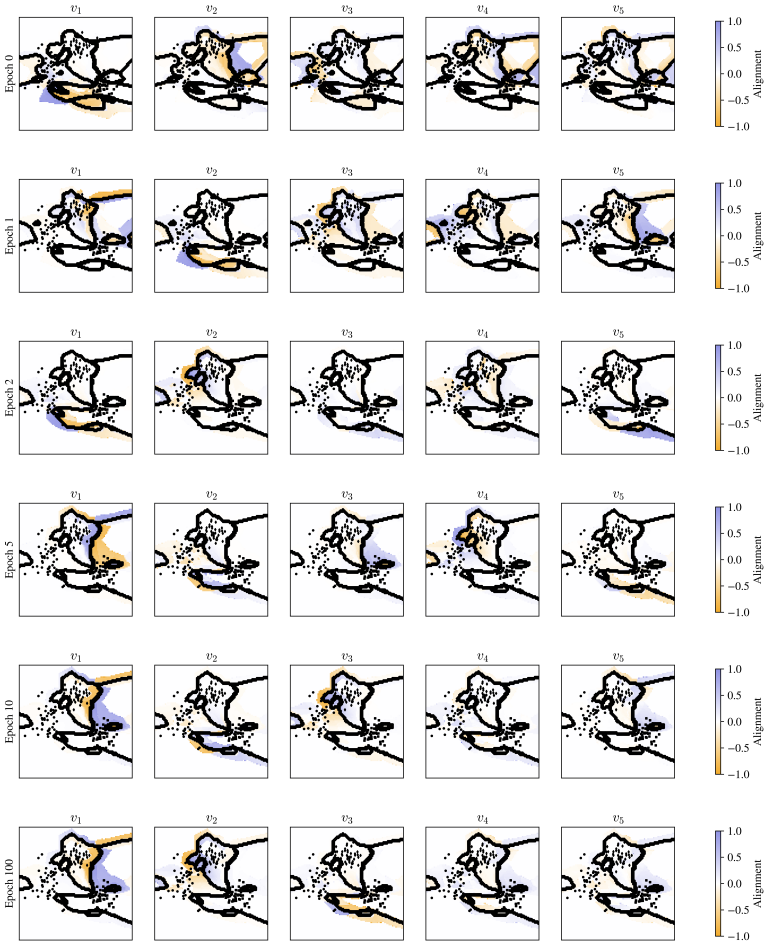

Figure 14: **Top Hessian eigenvectors encode boundaries also away from the minimum for an adversarial initialization on *gaussian*.** Epoch 0 is the boundary at initialization before training.

I Generalization measure and margin estimation technique

Going with the conventional wisdom that a simple decision boundary generalizes better than a complex one and the results from Section 2, we naturally define a generalization measure $\mathcal{G}_\theta$ that counts the number of eigenvectors needed to describe the decision boundary. Mathematically, we define $\mathcal{G}_\theta$ to be the ratio of Hessian eigenvectors with non-zero absolute mean alignment $\mathcal{A}$ with the training samples to the total number of eigenvectors, computed at the minimum θ:

$$m_i = \frac{1}{n}\sum_{s=1}^{n} |\mathcal{A}_i(x_s)| \quad ; \quad \mathcal{G}_\theta = \frac{1}{p}\sum_{i=1}^{p} \mathbb{1}[m_i > \epsilon], \tag{6}$$

where ϵ is close to zero,[3] $|\cdot|$ is the absolute value, and m_i denotes the mean of absolute alignment of the training samples with respect to eigenvector v_i. A better generalizing minimum has a smaller number of eigenvectors with a non-zero alignment of individual training data gradients, $\mathcal{G}_\theta$, signifying a simpler (therefore, better generalizing) decision boundary compared to other minima of the same network on the same data. In other words, there are fewer directions in the parameter space whose shift corresponds to large errors in the training data. Finally, as $\mathcal{G}_\theta$ depends crucially on training samples and the number of Hessian eigenvectors, the comparison of $\mathcal{G}_\theta$ between minima is meaningful only when they are reached with models with the same architecture and trained on the same data. Note that its value changes between training procedures with fixed hyperparameters due to randomness in the initialization.

I.1 Our generalization measure captures the complexity of the decision boundary

We compute our generalization measure $\mathcal{G}_\theta$ as in (6) for all simulated 2D datasets trained from normal, adversarial and large norm initializations, leading to decision boundaries of varied complexity and observe that $\mathcal{G}_\theta$ *captures the correct generalization order of the three minima* in all cases, as seen in Table 1. We also compare $\mathcal{G}_\theta$ with standard flatness measures such as the Hessian trace and its spectral norm. We confirm their superfluousness, as the Hessian trace and spectral norm of the normally initialized network with the simplest decision boundary are larger than for networks with adversarial and large norm initializations. Interestingly, the L_2 norm of the parameters also fails as a generalization measure despite the observation that well-generalizing solutions tend to have a minimum norm [41]. Those observations hold in the real datasets as presented in Appendix K.4.

Table 1: **Generalization measures comparison for the five simulated 2D datasets.** We provide the mean of those measures and their standard deviation over 5 runs. A bold font marks the best generalizing minimum according to the studied metric, green (red) color indicates whether the indication is correct (wrong). With yellow, we mark correct indications with standard deviations being larger than the difference of compared means.

Dataset	Training	$\mathcal{G}_\theta \downarrow$	$\mathrm{trace}(H) \downarrow$	$\lambda_{\max}(H) \downarrow$	$\|\|\theta^*\|\|_2 \downarrow$
gaussian	normal	**0.055** $\pm$ 0.004	0.176 $\pm$ 0.010	0.114 $\pm$ 0.010	**19.60** $\pm$ 0.15
	adversarial	0.156 $\pm$ 0.035	**0.003** $\pm$ 0.001	**0.002** $\pm$ 0.001	105.00 $\pm$ 0.005
	large norm	0.114 $\pm$ 0.040	0.021 $\pm$ 0.018	0.017 $\pm$ 0.012	98.169 $\pm$ 0.413
circle	normal	**0.044** $\pm$ 0.007	8.028 $\pm$ 0.777	4.965 $\pm$ 0.514	**22.884** $\pm$ 0.139
	adversarial	0.059 $\pm$ 0.003	**0.795** $\pm$ 0.051	**0.439** $\pm$ 0.037	41.630 $\pm$ 0.006
	large norm	0.057 $\pm$ 0.003	6.320 $\pm$ 0.833	4.350 $\pm$ 0.735	41.840 $\pm$ 0.226
half-moon	normal	**0.036** $\pm$ 0.003	4.202 $\pm$ 0.637	2.958 $\pm$ 0.479	**21.529** $\pm$ 0.285
	adversarial	0.072 $\pm$ 0.006	**0.037** $\pm$ 0.001	**0.017** $\pm$ 0.001	68.988 $\pm$ 0.009
	large norm	0.042 $\pm$ 0.004	1.119 $\pm$ 0.563	0.868 $\pm$ 0.431	64.807 $\pm$ 0.201
hierarchical	normal	**0.053** $\pm$ 0.001	12.450 $\pm$ 0.595	7.102 $\pm$ 0.189	**20.034** $\pm$ 0.202
	adversarial	0.118 $\pm$ 0.024	**3.394** $\pm$ 1.035	**2.645** $\pm$ 0.877	121.675 $\pm$ 0.062
	large norm	0.059 $\pm$ 0.009	93.104 $\pm$ 12.985	36.787 $\pm$ 3.936	112.579 $\pm$ 0.404
checkerboard	narrow-margin	0.046 $\pm$ 0.005	0.240 $\pm$ 0.050	0.127 $\pm$ 0.028	**19.267** $\pm$ 0.097
	wide-margin[a]	**0.043** $\pm$ 0.001	**0.029** $\pm$ 0.000	**0.014** $\pm$ 0.000	19.910 $\pm$ 0.000

[a] The standard deviation is almost 0 since we initialize models across runs with the same pretrained solution to promote a wide margin.

[3] To be precise, ϵ is set to a small number being the average maximum alignment with several random directions in a parameter space (see Appendix E). Usually, it is around 10^{-2}.

The proposed generalization measure also overcomes another weakness of the standard Hessian measures and is *invariant to reparametrization* as presented in Appendix J, as it relies on the decision boundary complexity which is also invariant to reparametrization. The $\mathcal{G}_\theta$ has also limitations. As it arises from the alignment of the gradients of loss of training samples with the Hessian eigenvectors encoding various sections of the boundary, it may fail to signal an increased complexity of the decision boundary far from any training sample, as we elaborate in the next subsection. The second limitation is related to the simplicity bias of neural networks, and we address it in Section I.3.

I.2 Limitations of $\mathcal{G}_\theta$

The $\mathcal{G}_\theta$ gives ambiguous results when distinguishing between the minima obtained with the normal and large norm training for *hierarchical gaussian*, marked in yellow in Table 1. While on average $\mathcal{G}_\theta$ successfully indicates that minima obtained with normal initialization have simpler decision boundaries than those obtained with the large norm initialization, the standard deviation exceeds the difference between the means. We identify a single case where our $\mathcal{G}_\theta$ fails to distinguish minima with different complexities of decision boundaries and make a full analysis of the alignment in Figures 15 and 16 for the *hierarchical gaussian* with four classes. Firstly, we see from the first column of Figure 15 that the complexity of the decision boundary increases in the region with a small number of training samples. It indicates a limitation of our measure that is based on the "interaction" between the training samples and the neighboring decision boundary. If the decision boundary is simple close to the training samples but complex away from them, $\mathcal{G}_\theta$ may struggle to detect this. Secondly, the number of outliers in the Hessian spectrum in the large norm case remains larger than in the normal case, as visible in the last column of Figure 15. Finally, we take a closer look at the alignment of the top Hessian eigenvectors and gradients of loss of training samples in Figure 16 at the minima studied in Figure 15. We still see that for simpler decision boundaries, the alignment of the training gradients localizes much more in the top Hessian subspace than for the complex boundaries. At the same time, $\mathcal{G}_\theta$ is almost the same for both cases, which means that it is an imperfect measure of the localization of the gradient alignment that seems to be a prevailing characteristic of minima with simple decision boundaries. We leave the improvement of this scalar measure for further study. At the same time, we stress that $\mathcal{G}_\theta$ has correctly distinguished between minima with simple and complex decision boundaries from adversarial initializations in all the studied cases, and the ambiguity arises only in the large norm initializations.

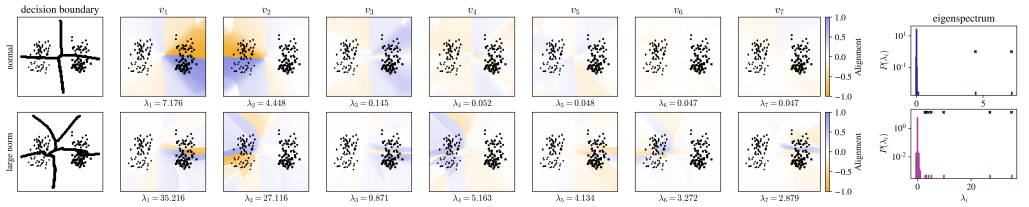

Figure 15: **Decision boundaries of different complexities for *hierarchical gaussian*.** Alignment plots and histograms of the Hessian spectra for models obtained from normal training and a large norm initialization.

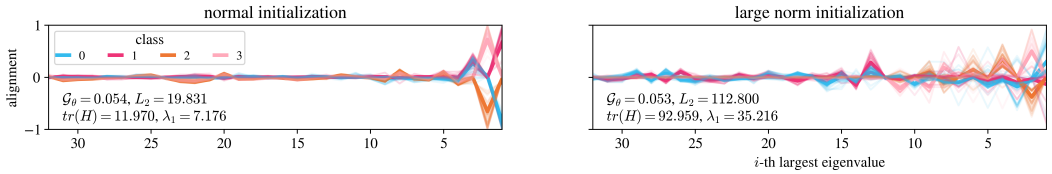

Figure 16: **Alignment of all training data with the top** 25 **eigenvectors for *hierarchical gaussian*** for models obtained from the *(Left)* normal training and the*(Right)* large norm initialization. There are four classes $\{0, 1, 2, 3\}$. The dark lines show the mean of each class alignment.

The second limitation of the proposed measure arises when the network exhibits a simiplicity bias. Simplicity bias is the tendency of neural networks to learn "simple" models and has been

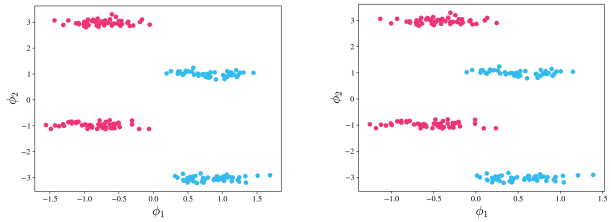

Figure 17: **The *checkerboard* datasets** used for *(Left)* training and *(Right)* pretraining to achieve the wide-margin solution.

hypothesized to explain the generalization properties of neural networks [4, 30]. Shah et al. [39] claims that simplicity bias may instead hurt generalization (in SGD and its variants), when networks prefer simpler features over complex ones that are more informative for prediction. In the context of the decision boundary, the simplicity bias is related to a bias towards a more linear boundary. *As our generalization metric measures the simplicity of the decision boundary, it may fail to signal when the generalization capability of a model is diminished by simplicity bias.*

To validate it, we use the synthetic *checkerboard* dataset of Gaussian clusters, similar to the one analyzed by Shah et al. [39]; it has two classes and two features – one feature is simple (single linear boundary sufficient for 100% accuracy), and the other is complex (100% prediction needs at least $n-1$ linear pieces for n clusters). It is presented in the left column of Figure 17. We study the minima reached by models trained to classify this dataset in two settings. The first setting is normal initialization, affected by simplicity bias, resulting in a more linear boundary and a narrow margin (see the first row of Figure 18). In the second setting, which we call *wide-margin*, we encourage a boundary with a wider margin by pretraining on another dataset presented in the right column of Figure 17, and then training on the same *checkerboard* dataset as in the first setting.

In Figure 18, we show the decision boundary and the alignment of gradients in input space with the top three eigenvectors v_i. The Hessian eigenspectrum, presented in the fifth column, exhibits two outliers for both initializations. While the second setting has a wider margin and thus exhibits better generalization, the difference between $\mathcal{G}_\theta$ for normal and wide-margin initializations is small. To properly distinguish those solutions, we can instead estimate the margin width of the decision boundary as described in the next section.

I.3 The order of eigenvectors is related to the margin width of the decision boundary

We observe in every setup (Figure 1 and Appendix B) that *the order of the top eigenvectors follows the increasing margin of the encoded sections of the boundary*. The topmost eigenvector captures the boundary section that separates the closest training data from two classes in the input space.

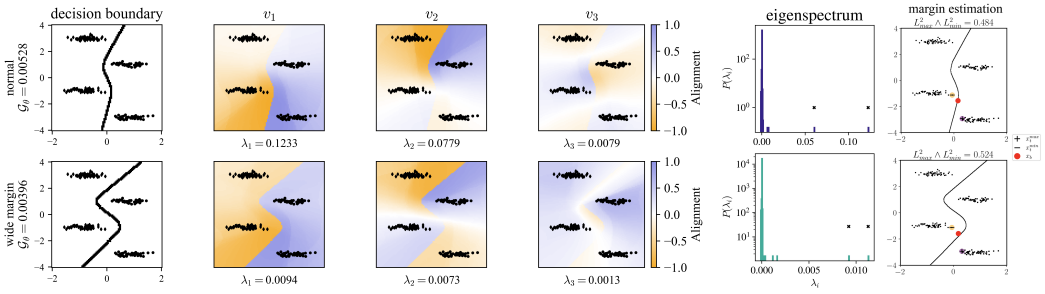

Figure 18: **Simplicity bias and margin estimation for *checkerboard*.** We compare the alignments of the top three eigenvectors v_i and the eigenspectrum, trained using two different initializations, normal *(top row)* and wide-margin initialization *(bottom row)*. Both initializations have very similar generalization measures. *(Last column)* Margin estimation from $x_b, x_i^{\max}$ and $x_i^{\min}$.

This opens up a possibility to *estimate the margin of the decision boundary in higher input dimensions.* To do so, we need two data points from the input space: a training sample x_t that is closest to the boundary (the one that determines the smallest margin of the decision boundary) and a sample on the boundary x_b, which should be as close to x_t as possible, as this determines how good estimate of the margin we have. x_t is chosen to have the largest alignment with the top Hessian eigenvector v_1. Note that, a priori, we do not know the alignment sign of the training data x_t closest to the artificial sample on the boundary x_b. Hence, x_t^{min} and x_t^{max} are the smallest and largest alignments with v_1 (yellow and purple dots in the last column of Figure 18). To find the sample on the boundary (red dots in Figure 18), we can optimize the features of an input sample such that its gradient has a maximum alignment with the top Hessian eigenvector. Then we compute the L_2 distance between x_b and x_t^{min}, and between x_b and x_t^{max} in the input space and choose the smaller L_2 distance as the margin. An example of such a margin width estimation is presented in Figure 18, where we correctly see that the less linear decision boundary corresponds to a wider margin compared to the linear decision boundary. This simple yet effective margin estimation technique, together with our generalization measure $\mathcal{G}_\theta$, enables a better understanding of the generalization ability of deep neural networks.

J Generalization measure is invariant to model reparameterization

A natural assumption is that if a model after a reparameterization yields the same output as the original one, their generalization abilities (and measures) should also be equal. This is not the case for metrics based on Hessian trace and ReLU networks; One can "artificially" sharpen a minimum while retaining the predictions of the original model using the α-scale transformation proposed by Dinh et al. [10]. In Figure 19, we see that while such a reparameterization indeed affects the Hessian spectrum, it does not impact the connection between the top Hessian eigenvectors and decision boundary. As a result, *our generalization metric is also invariant to the reparameterization* as it is based on the simplicity of the decision boundary that stays the same. At the same time, we see that the reparameterization may change the sign and values of the alignment. It is yet unclear why this happens, but this may further suggest that the exact values of the alignment are not informative.

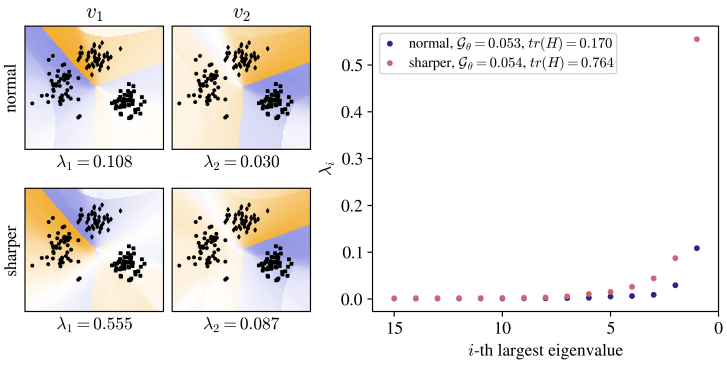

Figure 19: **Reparameterization.** A sharp α-scale transformation for a 2-layer ReLU-network that rescales the weights and biases of the original model according to [10, App. B] while keeping the predictions identical. *(Left)* The alignments for the top eigenvectors. *(Right)* The spectra for both parameterizations.

K Validation of the observations on real datasets: *Iris*, *MNIST*, and *CIFAR-10*

We have conducted our Hessian analysis for 2D datasets enabling straightforward visualization of the learned decision boundary. This approach has enabled a clear visual distinction between simple or complex decision boundaries. Such a visual distinction is much needed in view of a limited (to our knowledge) theoretical description of the complexity of the decision boundary. At the same time, visualization of the decision boundary is hardly possible for high-dimensional datasets.

Here, we extend our analysis to real datasets, that is to *Iris* dataset in section K.1, *MNIST*-based datasets in section K.2, and *CIFAR-10*-based datasets in section K.3. Above all, we show that *with our Hessian analysis, we distinguish between well- and badly generalizing minima in realistic deep learning setups*. To do so, we compare the models trained with a regular initialization and the adversarial initialization [27], which are believed to reach a well- and badly generalizing minimum, respectively. In section K.4, we showcase that the proposed generalization measure G_θ *correctly distinguishes well- and badly-generalizing minima* in the case of *Iris* dataset and *MNIST*-based datasets.

K.1 Hessian analysis for *Iris*

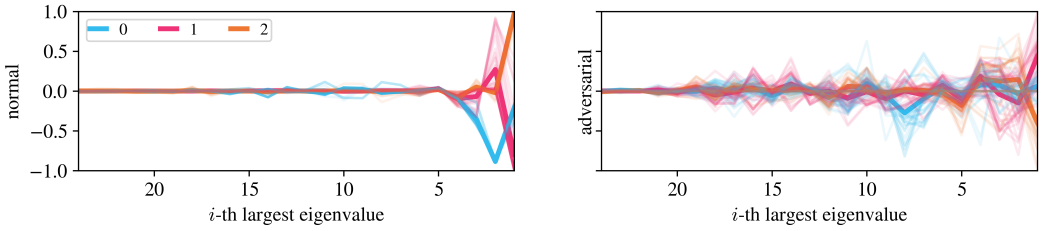

Figure 20: **Alignment of all training data with the top** 25 **Hessian eigenvectors for *Iris*.** *(Left)* Normal training. *(Right)* Adversarial initialization. There are three classes $\{0, 1, 2\}$. The dark lines show the mean of each class alignment.

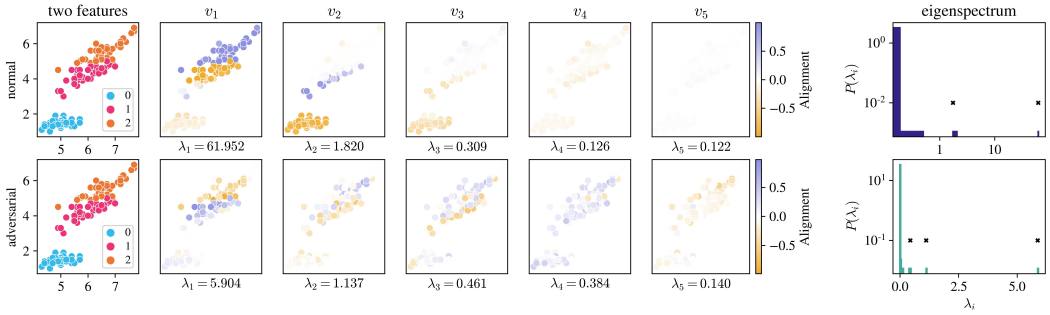

Figure 21: **Experimental results on *Iris*.** *(First column)* Two features (petal and sepal length) out of four of the Iris dataset with color-coded classes. *(Other columns)* The alignment of gradients of the loss of individual training samples with the top five Hessian eigenvectors. *(Last column)* Histograms of the Hessian spectra. *(Top)* Well-generalizing minimum obtained with normal training. *(Bottom)* Badly generalizing minimum obtained with an adversarial initialization [27].

For *Iris*, we show the corresponding Hessian spectra in the last column of Figure 21 and alignments of gradients of loss of individual training samples with the Hessian eigenvectors in Figure 20, respectively. We again see the larger number of outliers in the spectra in the case of more complex decision boundary. Most importantly, we see that the gradients have non-zero alignment with a much smaller number of Hessian eigenvectors in the case of normal training than in the adversarial case (Figure 20). We again see that the gradients are more aligned with each other in the well generalizing than badly generalizing minimum, as observed during the training dynamics in Chatterjee & Zielinski

[6]. The generalization metric $\mathcal{G}_\theta$ captures this difference as expected (is lower for well generalizing minimum) and is listed along with the *MNIST* results in Table 2 in Appendix K.4.

Moreover, our low-dimensional analysis in the main body shows that the drastically different behavior of training gradients alignment with the Hessian eigenvectors results from a different complexity of the decision boundary. In other words, training gradients align with a larger number of the top Hessian eigenvectors because around training samples in input space, there are numerous sections of the decision boundary encoded in multiple directions in parameter space. While we could make this connection clear in the case of 2D datasets, it is more challenging for four dimensions and impractical for significantly larger dimensions. For Irises, we instead visualize samples by selecting only two features out of four and without the decision boundaries. Then we color code the alignment of the gradient of their individual losses at the minimum with the top Hessian eigenvectors. We present the normal and adversarial training results in Figure 21. We can see a clearly different behavior of the alignment between the well- and badly-generalizing minimum. This suggests a different complexity of the decision boundary following results from the low-dimensional data.

K.2 Hessian analysis for *MNIST*

Finally, we make an analogous Hessian analysis for the *MNIST*-based datasets. To decrease the complexity of the dataset and better understand the dependence of the results on the number of classes, we create four subsets of *MNIST*: *MNIST-017*, *MNIST-179*, *MNIST-0179*, and *MNIST-1379*, where numbers indicate selected classes of digits. Each class has a few hundred samples sampled randomly from the *MNIST* dataset.

The analysis of the alignment of the gradients of loss of individual training samples and the top Hessian eigenvectors is presented in Figure 22. We continue to see that the alignment for the regular training is more localized in the space spanned by the top Hessian eigenvectors. We also see self-alignment of the gradients [6] that maybe stops being so apparent in the top few eigenvectors.

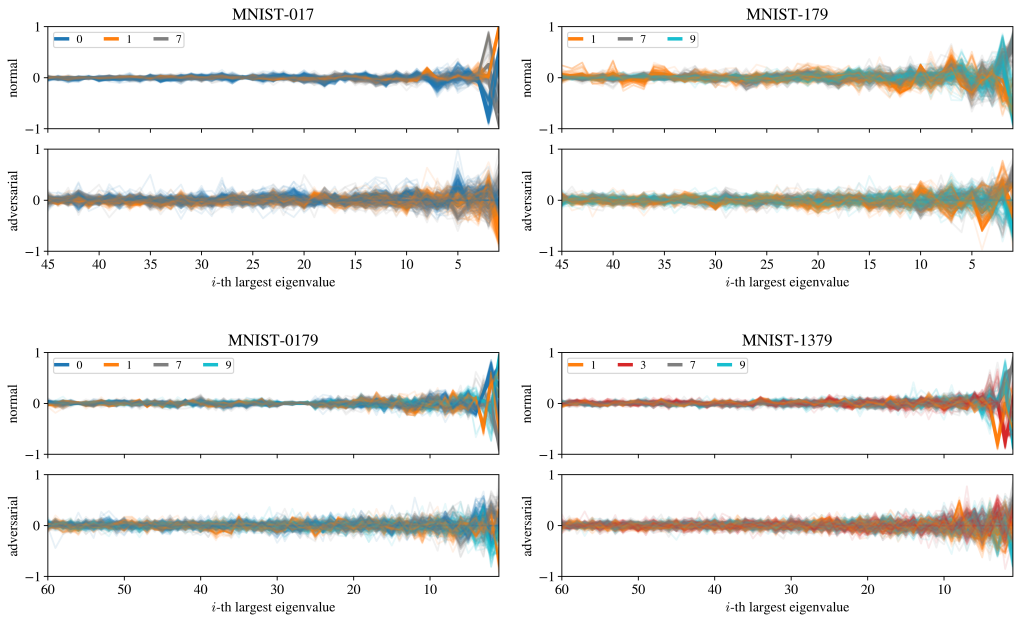

Figure 22: **Normal and adversarial initialization training for *MNIST-017*, *MNIST-179*, *MNIST-0179*, and *MNIST-1379***. We plot the alignments of gradients of all training samples with all eigenvectors ordered by their eigenvalues. Only the largest eigenvectors have non-zero alignment with the gradients of training samples, and their number increases for the training from the adversarial initialization.

Moreover, as we mentioned in Appendix K.1, while we could make a clear connection between Hessian-gradient alignment and complexity of decision boundary in the case of 2D datasets, such a

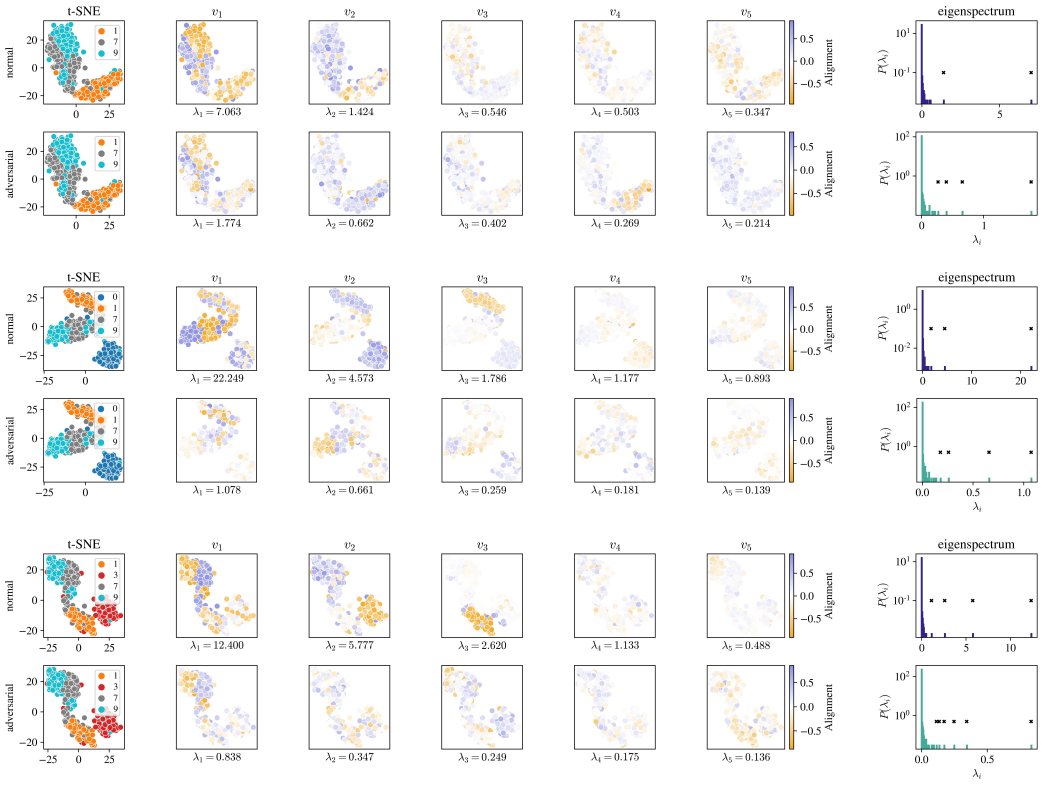

Figure 23: **Normal and adversarial initializations training for *MNIST-179*, *MNIST-0179*, and *MNIST-1379* with t-SNE visualization and Hessian eigenspectra**. We visualize the *MNIST*-based datasets with t-SNE and color code the alignments of gradients of all training samples onto all eigenvectors ordered by their eigenvalues. Multiple eigenvectors have non-zero alignment with the gradients of training samples, and there is little ordering of the samples' colors suggesting complex decision boundaries. *(Last column)* Hessian eigenspectra.

visualization is impractical for high input dimensions. Instead, we make the following non-rigorous analysis. We visualize the high-dimensional *MNIST* samples in a 2D plot using t-distributed stochastic neighbor embedding (t-SNE) and then color code the alignment of the gradient of their individual losses at the minimum. We present the normal and adversarial training results in Figure 23. Even if there is no guarantee that the neural network representation of the data is related to the one obtained by t-SNE nor that the learned decision boundary in the input space corresponds simply to the boundaries between t-SNE generated clusters, we still can see a clearly different behavior of the alignment between the well- and badly-generalizing minimum.

Finally, the Hessian spectra for all *MNIST*-based datasets are in the last column of Figure 23. We consistently see that the number of outliers increases for the badly generalizing minima. We also see that metrics like the Hessian trace or its largest eigenvalue fail to capture the difference between the minima's generalizing abilities. On the other hand, our generalization measure, $\mathcal{G}_\theta$, consistently provides correct indications. We make this comparison apparent in Table 2 in Appendix K.4.

K.3 Hessian analysis for CIFAR-127

We make identical observations for a *CIFAR-10*-based dataset, namely on *CIFAR-127*, where digits indicate classes taken for analysis. This time, we used LeNet-5 for the classification task. Ther results are presented in Figures 24 and 25.

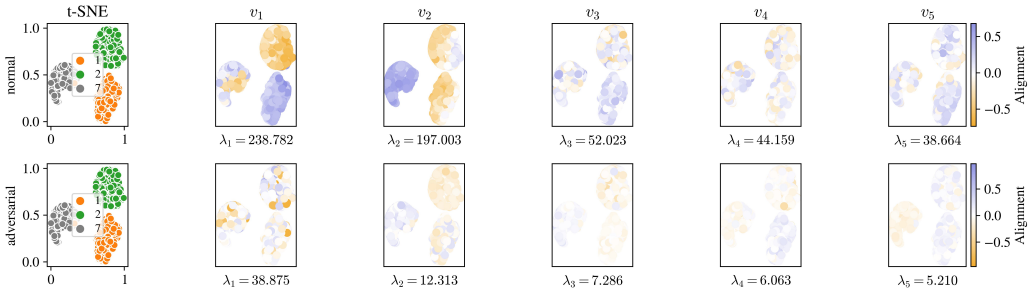

Figure 24: **Normal and adversarial initializations training for three selected classes of *CIFAR-10* and LeNet-5**. We visualize the *CIFAR-10*-based dataset with t-SNE and color code the alignments of gradients of loss of all training samples onto the top five eigenvectors ordered by their eigenvalues. In the first row, the normal training leads to simpler boundaries indicated by the alignment localized in the top two Hessian eigenvectors. In the second row (adversarial initialization), multiple eigenvectors have non-zero alignment with the gradients of training samples, and there is little ordering of the samples' colors suggesting complex decision boundaries. We computed the top eigenvalues and eigenvectors of the Hessian using the iterative power method [16].

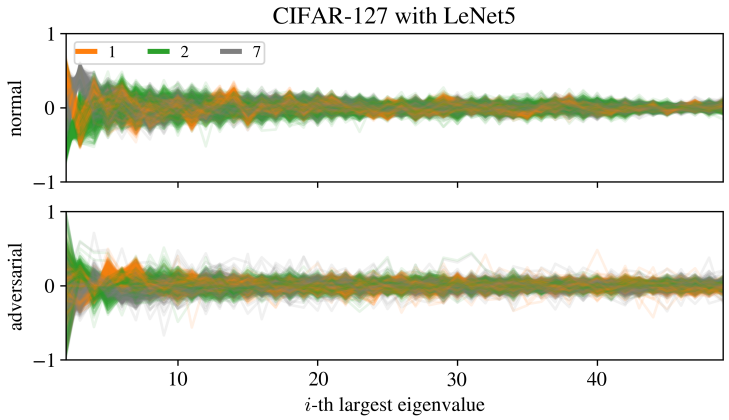

Figure 25: **Normal and adversarial initialization training for three classes of *CIFAR-10* and LeNet-5**. We plot the alignments of gradients of all training samples onto the top 50 eigenvectors ordered by their eigenvalues. In the case of adversarial initialization (implying the complex decision boundary), training samples keep having a significant non-zero alignment with the top Hessian eigenvectors far into the spectrum. We computed the top eigenvalues and eigenvectors of the Hessian using the iterative power method [16].

Table 2: **Generalization measure comparison for real datasets *Iris* and different subsets of *MNIST* under different initializations.** We provide the mean of those measures and their standard deviation over 5 runs. A bold font marks the best generalizing minimum according to the studied metric, green (red) color indicates whether the indication is correct (wrong).

Dataset	Training	$\mathcal{G}_\theta \downarrow$	$\mathrm{trace}(H) \downarrow$	$\lambda_{\max}(H) \downarrow$	$\|\|\theta^*\|\|_2 \downarrow$
Iris	normal	**0.031**±0.006	67.857±5.943	65.005±6.072	**13.998**±0.221
	adversarial	0.094±0.002	**8.324**±0.235	**5.934**±0.067	68.361±2.040
MNIST-017	normal	**0.037**±0.028	6.288±4.697	3.758±3.215	2237.6±3526.8
	adversarial	0.109±0.002	**0.945**±0.117	**0.479**±0.097	**938.38**±0.03
MNIST-179	normal	**0.045**±0.063	14.714±14.783	8.270±8.262	731.65±716.37
	adversarial	0.209±0.006	**5.222**±0.740	**1.611**±0.271	**268.65**±0.06
MNIST-0179	normal	**0.043**±0.010	28.472±9.840	13.254±6.312	387.54±129.66
	adversarial	0.110±0.003	**3.180**±0.399	**0.946**±0.203	**209.41**±0.01
MNIST-1379	normal	**0.077**±0.031	18.380±16.588	6.868±6.368	**249.63**±135.32
	adversarial	0.135±0.015	**2.938**±0.164	**0.844**±0.029	398.19±0.17

K.4 Generalization measure for *Iris* and *MNIST*

Finally, in Table 2, we present values of the generalization metric $\mathcal{G}_\theta$ introduced in Equation 6 for models trained on various real datasets and from different initializations. "Normal training" indicates the regular initialization of the neural networks. Adversarial initialization follows the initialization by Liu et al. [27] that consists in pretraining on the same data but with random labels, as discussed in Section 3. In the low-dimensional datasets, we consistently see that models initialized adversarially learn more complex decision boundaries that generalize worse than the simple boundary learned by regularly trained models. As we cannot visualize the decision boundary for the high-dimensional data, we skip the analysis of the large-norm initialization here. Instead, we use only the established adversarial initialization by Liu et al. [27], which has been shown to produce complex boundaries and badly generalizing minima. The generalization metric successfully distinguishes between those models. The analogous results for simulated 2D datasets are in Table 1.

L Discussion and related works

Properties of the Hessian. Our work sheds light on various universal aspects of the training loss Hessian. While the localization of the gradient information in the top subspace of the Hessian is known [18], the observation that the top Hessian eigenvectors align with the gradients of samples at the decision boundary, therefore, they encode the decision boundary, provides a new perspective and a simpler way to study the complexity of high-dimensional decision boundary. In fact, our work aligns with the mathematical analysis of Papyan [32] who connected the emergence of outliers (and therefore the top Hessian eigenvectors) to the "between-class gradient second moment". Moreover, our work illustrates that the number of outliers in the Hessian spectrum is related to the complexity of the learned decision boundary, which was so far rather connected to the number of classes. Our findings shed light on the observation of Jastrzebski et al. [22] that the loss in the subspace of the top Hessian eigenvectors is "bowl-like" and that decreasing learning rate within this subspace leads to better generalizing solutions: this procedure may lead to maximizing the margin of the decision boundary.

Generalization. We confirm that the Hessian-based metrics like the trace or the largest eigenvalue are unreliable proxies for measuring the generalization ability of deep learning models. Instead, these metrics are more dependent on the training parameters like the number of epochs, as observed in Sagun et al. [37]. For example, Table 1 shows that the Hessian trace is the smallest for badly generalizing minima, which required the largest number of epochs to train. Interestingly, we see that within-class gradients at the minimum are more aligned with each other in the better generalizing case, as noted in Chatterjee & Zielinski [6]. Our findings suggest that this gradient "coherence" emerges from the simplicity of the decision boundary and disappears as the complexity increases.

Robustness and interpretability. The robustness of neural networks is an active area of research aiming to produce stable outputs towards small, semantically irrelevant input perturbations. Feng & Tu [13] mapped the variations in inputs to variations of specific weight parameters, through which the Hessian connection between the input and parameter space is established. Combining this with our work may lead to a better understanding of why Hessian-based techniques could lead to more robust models [29, 36, 45, 40] or more robust gradient-based explanations Dombrowski et al. [11, 12]. In fact, they may simplify the decision boundary, causing gradients of similar inputs to align in one direction. This also explain the success of Hessian-based interpretation of neural networks [25, 28, 9] and in pruning [26, 43].

Practical implications. Establishing the connection between Hessian and the decision boundary learned by the network provides a new tool to study the boundary in high input dimensions. However, exact computation of the Hessian and its spectrum is hard for both large data and models. Instead, we can follow the observation from Appendix M and Ghorbani et al. [15] that the top eigenvectors of the Hessian at the minimum have a large overlap with the top eigenvectors of the gradient covariance matrix, which enables efficient alignment computation but still requires an expensive eigendecomposition. We can also use efficient approximation techniques based on the Hessian-vector product [34, 1, 16] and the generalized Gauss-Newton decomposition of the Hessian [37, 31, 32].

M The gradient covariance matrix vs the Hessian at the minimum

Here, we study the covariance matrix of gradients of loss of individual training samples at the minimum defined as

$$\begin{aligned}\Sigma(\theta;\mathcal{D}) &= \frac{1}{n}\sum_{i=1}^{n} g_\theta(x_i) g_\theta^T(x_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial\theta}\mathcal{L}\left(\theta;\{x_i,y_i\}\right)\frac{\partial}{\partial\theta}\mathcal{L}^T\left(\theta;\{x_i,y_i\}\right)\,, \end{aligned} \tag{7}$$

where training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $x_i \in \mathbb{R}^d$, and $y_i \in \{1, \ldots, C\}$ is the class label. In Figure 26, we show that the top few eigenvectors of the covariance matrix $\Sigma(\theta;\mathcal{D})$ actually encode the same information as the top few Hessian eigenvectors at the minimum as observed by Ghorbani et al. [15] and Fort & Ganguli [14]. This observation can also be justified by the Hessian approximation with the gradient outer product holds well at the minimum. Therefore, *the top subspace of* $\Sigma(\theta;\mathcal{D})$ *can be used instead of the more computationally expensive Hessian to study the decision boundary.*

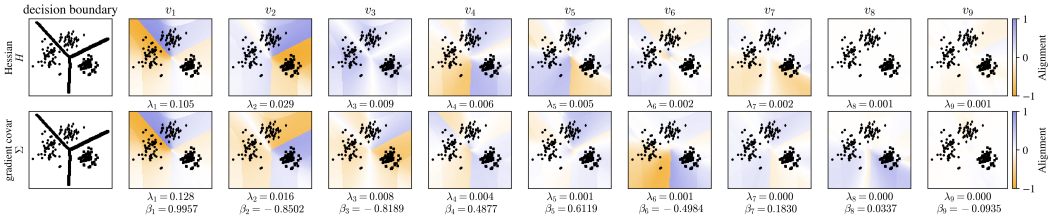

Figure 26: **Gradient covariance matrix vs. Hessian.** We compare the alignment of gradients of loss of input samples with the top eigenvectors of *(Top)* the Hessian H and *(Bottom)* the gradient covariance matrix $\Sigma(\theta;\mathcal{D})$ defined in Equation 7. For the first two eigenvectors of both matrices, their alignment with gradients of input samples is very similar across the input space. The cosine similarity $\beta_i = \langle v_i^H, v_i^\Sigma \rangle$ of between the Hessian eigenvectors v_i^H and the gradient covariance matrix' eigenvectors v_i^C decreases with the values of their eigenvalues.