On Generalization of Spectral Gradient Descent: A Case Study on Imbalanced Data

Bhavya Vasudeva* BVASUDEV@USC.EDU
University of Southern California, Los Angeles, USA

Puneesh Deora* PUNEESHDEORA@ECE.UBC.CA
University of British Columbia, Vancouver, Canada

Christos Thrampoulidis CTHRAMPO@ECE.UBC.CA
University of British Columbia, Vancouver, Canada

Abstract

The growing adoption of spectrum-aware matrix-valued optimizers such as Shampoo and Muon in deep learning motivates a systematic study of their generalization properties and, in particular, when they might outperform competitive methods. We approach this challenging question by introducing appropriate simplifying abstractions as follows: First, we use imbalanced data as a testbed for studying the behavior of spectrum-aware optimizers. Second, we study the canonical form of such optimizers, which is Spectral Gradient Descent (SpecGD)—each update step is $\boldsymbol{U}\boldsymbol{V}^T$ where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the truncated SVD of the gradient. Third, within this framework we identify a minimal linear setting where we can analyze when SpecGD outperforms vanilla GD. We show that unlike GD, which prioritizes learning majority classes first, SpecGD initially learns all principal components of the data at equal rates. We demonstrate how this translates to a growing gap in balanced accuracy favoring SpecGD early in training.

1. Introduction

Spectrum-aware optimizers such as Shampoo [8] and Muon [10] have recently gained significant traction in the deep learning community, delivering substantial training speedups for deep classifiers [10] and transformer language models [13, 20] compared to standard practices like SGD [15] with momentum or Adam [11]. The key distinction lies in how these methods treat neural network parameters: while SGD and Adam operate entry-wise on vectorized parameters, Shampoo and Muon work directly with matrix-valued parameters (such as weight matrices and attention matrices) at the layer level. This matrix-level approach intuitively enables optimization trajectories that entry-wise methods cannot achieve. Despite their empirical success, a fundamental question remains unanswered: **when do spectrum-aware optimizers generalize better than standard methods?**

The challenge is substantial. Even well-established optimizers like Adam, despite over a decade of practical dominance, remain poorly understood compared to SGD, whose Euclidean gradient descent trajectory is well-characterized both statistically and algorithmically. Recent theoretical progress has begun to illuminate these methods through the lens of implicit bias. For instance, while the implicit bias of gradient descent toward max-margin classifiers with respect to the ℓ_2 norm has been established [9, 19], recent work proved that Adam converges to a max-margin classifier with respect to the ℓ_∞ norm in linear settings [21]. Conceptually, this difference can be understood by

*Equal contribution

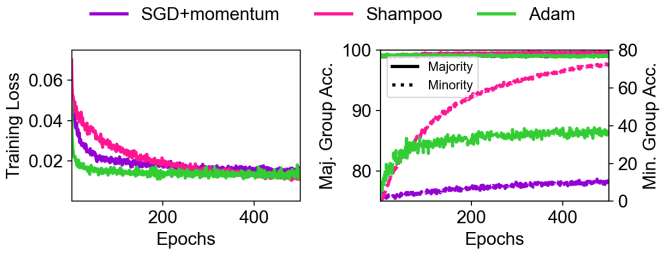

Figure 1: Results for training a one-hidden-layer MLP CMNIST dataset with 99% digit–colour correlation (see Section 2 for details) using SGD with momentum, Shampoo and Adam. All three optimizers trained to comparable train loss achieve near-perfect test accuracy on the majority groups (same digit and color labels) but SGD and Adam have much lower test accuracy on the minority groups (opposite digit and color labels) as compared to Shampoo.

realizing that Adam reduces to SignGD when momentum and preconditioning histories are ignored (*i.e.*, β parameters set to zero) [6]. This reduction has been leveraged before to argue about Adam properties, because often SignGD is simpler to analyze, e.g., Kunstner et al. [12].

A similar reduction exists for spectrum-aware optimizers: Shampoo without preconditioning history and Muon with perfect matrix operations reduce to Spectral Gradient Descent (SpecGD) [3]. Thus, just as SignGD serves as the canonical form for understanding Adam, SpecGD might provide the key to understanding Shampoo and Muon. Both SignGD and SpecGD are instances of normalized steepest descent with respect to ℓ_∞ and spectral norms, respectively [2, 6].

Recent work Fan et al. [6] characterized SpecGD's optimization trajectory in linear multi-class classification, showing that its implicit bias drives weights toward a max-margin classifier with respect to the spectral norm. However, these results have two limitations. First, they only describe the algorithm's behavior in the terminal phase of training, which may not reflect practical deep learning scenarios that often employ early stopping. Second, and more important, implicit bias results provide no direct guarantees about generalization performance—the ultimate objective in machine learning.

Motivated by the apparent lack of understanding generalization properties of SpecGD, as a starting point, we ask: **Can we identify concrete settings where SpecGD generalizes better than standard (Euclidean) GD?** We focus on minimal settings for two reasons: (a) understanding benefits in simple cases can demystify often contradictory performance reports in large-scale models, and (b) minimal settings are more amenable to theoretical analysis that can formalize our intuitions.

1.1. Contribution

Imbalanced data as playground. We introduce imbalanced data as a testbed for studying SpecGD's potential generalization advantages. Fig. 1 provides a concrete demonstration: we train a one-hidden-layer MLP on Colored-MNIST under severe group imbalance, where each digit appears in its majority color 99% of the time during training, creating a strong but spurious digit-color correlation. At test time, we evaluate on majority and minority group with same and opposite digit-color associations, respectively. The results are revealing: whereas SGD (with momentum) and Adam overfit the spurious correlation and suffer a marked drop in accuracy on the minority group, Shampoo maintains high performance suggesting that spectrum-aware updates can curb reliance on

spurious features. This poses a natural question: *Under which kinds of data imbalance and training regimes do spectrum-aware optimizers provably outperform ordinary gradient descent?*

Minimal setting. We pursue this question in the simplest setting that still captures the key tension illustrated in Fig. 1: a linear classifier trained with a squared-loss objective under class imbalance. Although the Colored-MNIST experiment involves a richer group imbalance (digits × colors), its difficulty ultimately stems from the same mechanism—an under-representation of certain label values and the temptation to fit easy but spurious correlations. By collapsing the color dimension and focusing on label imbalance alone, we obtain a tractable model that lets us isolate and analyse the implicit regularisation imposed by different update rules.

Theoretical comparison on class-imbalanced data. For a linear model, we derive closed-form expressions for the training trajectories of Euclidean GD and Spectral GD (Shampoo without accumulation). We show that, with early stopping, Spectral GD achieves a lower balanced-class risk than GD, whereas both approach the same risk asymptotically in time.

2. Results on Colored-MNIST

As discussed in Section 1.1, we first compare three optimizers—SGD with momentum, Adam and Shampoo—on a variant of the Colored-MNIST (CMNIST) dataset, a benchmark used commonly in the literature on spurious correlations [1, 14, 16]. The task is to classify each digit as either < 5 or ≥ 5. The digits in each class are injected with a background color (red or green) that is correlated with the label and acts as a spurious feature. In our setting, the spurious correlation in the train set is 99%. To assess the reliance of the trained model on the color or the digit features, we evaluate test accuracies on two groups: the majority group (samples where the spurious feature label and the class label are the same) and the minority group (samples where they differ).

We train a one-hidden-layer ReLU MLP with fixed final layer weights (see App. D for details). Fig. 1 compares the train loss and test accuracies on majority and minority groups for the three optimizers. While all optimizers reach low training loss and high accuracy on the majority group, Shampoo attains much higher accuracy on the minority group. To demystify this behavior and analyze the implicit regularization of different update rules, we next transition to a simplified setting.

3. Linear Model on Class-Imbalanced Data

Here, we train a linear model with the canonical forms of the three optimizers considered in the previous section (*i.e.*, NGD, SpecGD and SignGD), on class imbalanced data. This as a minimal setting that retains key features of the CMNIST experiment, and is amenable to theoretical analysis.

3.1. Notations and Algorithms

We denote matrices, vectors and scalars by $\boldsymbol{A}$, $\boldsymbol{a}$, and a, respectively. We denote the (i, j)-th entry of matrix $\boldsymbol{A}$ as $A[i, j]$. Let $\|\cdot\|_F$, $\|\cdot\|_2$, and $\|\cdot\|_{\max}$ denote the Frobenius, spectral and $\max$ norms, respectively, where $\|\boldsymbol{A}\|_{\max} := \max_{i,j} A[i, j]$. Let $\|\boldsymbol{a}\|$ denote the ℓ_2 norm of $\boldsymbol{a}$. $\mathbb{1}[\cdot]$ denotes the indicator function, *e.g.*, $\mathbb{1}[a \geq b] = 1$ if $a \geq b$ and 0 otherwise.

Let $\boldsymbol{W} \in \mathbb{R}^{K\times d}$ denote the weight matrix of a linear model, and $\mathcal{L}(\boldsymbol{W})$ denote the loss function. Let $\boldsymbol{W}_t$ and $\boldsymbol{\nabla}_t := \nabla\mathcal{L}(\boldsymbol{W}_t)$ denote the iterate and gradient at time t, respectively. The updates for normalized steepest descent with respect to norm $\|\cdot\|$, with step-size $\eta > 0$, are [4]:

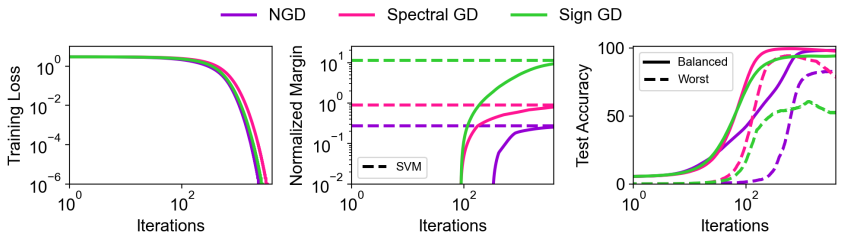

Figure 2: Results for training a linear model with NGD, SpecGD and SignGD on heavy-tailed class-imbalance data using cross-entropy loss. Early-stopped SpecGD achieves higher class-balanced and worst-class accuracy compared to other update rules or stopping points.

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \boldsymbol{\Delta}_t, \text{ where } \boldsymbol{\Delta}_t := \operatorname{argmax}_{\|\boldsymbol{\Delta}\| \leq 1} \langle \boldsymbol{\nabla}_t, \boldsymbol{\Delta} \rangle. \tag{1}$$

As discussed in Section 1, NGD, SpecGD and SignGD are instances of normalized steepest descent Eq. (1) with respect to Frobenius, spectral and $\max$ norms, respectively (also see Appendix A).

3.2. Data Model

Let $y \in \{1, \ldots, k\}$ denote the class labels, and let the corresponding one-hot labels be denoted as $\boldsymbol{y} \in \{\boldsymbol{e}_c\}_{c=1}^k$, where $\boldsymbol{e}_c$ is the c-th standard basis vector in $\mathbb{R}^k$. Class probabilities are given by $p_c := \Pr(y = c)$ such that $\sum_{c=1}^k p_c = 1$. Each class c has an associated mean vector $\boldsymbol{\mu}_c \in \mathbb{R}^d$, and samples for class c are generated as isotropic Gaussians with mean $\boldsymbol{\mu}_c$. Finally, we assume that the means $\boldsymbol{\mu}_c$ are orthonormal. Put together, the data model we study is such that:

$$\Pr(y = c) = p_c, \; c \in [k], \qquad \boldsymbol{x}|y \sim \mathcal{N}\big(\boldsymbol{\mu}_y, \sigma_x^2 \mathbb{I}_d\big), \qquad \text{and} \quad \boldsymbol{\mu}_1 \perp \ldots \perp \boldsymbol{\mu}_k \,, \|\boldsymbol{\mu}_c\| = 1. \tag{DM}$$

3.3. Experimental Results

For the experiments, we consider a heavy-tailed class imbalance setting by choosing $p_c \propto \frac{1}{c}$, and we sample each $\boldsymbol{\mu}_c$ independently from a zero-mean isotropic Gaussian distribution and normalize it. We use 20 classes, 100 samples, $d = 200$ and $\sigma_x = 0.1$. We initialize $\boldsymbol{W}_0$ by sampling each entry independently from $\mathcal{N}(0, \frac{1}{d})$. We use learning rates 0.025, 0.005 and 5×10^{-4} for NGD, SpecGD and SignGD, respectively. These choices are made such that train loss curves of the algorithms are comparable.

Fig. 2 shows the results for training a linear model in this setting using NGD, SpecGD, or SignGD, to minimize the cross-entropy loss. We observe that for all three update rules, as train loss approaches 0, the model converges to a solution that maximizes the margin defined with respect to the corresponding norm. This long-training behavior was recently shown by Fan et al. [6]. However, comparing the test performance, we find that early-stopped SpecGD attains higher class-balanced and worst-class test accuracies compared to NGD or SignGD at any stopping point.

3.4. Theoretical Analysis

In this section, we analyze and compare the dynamics of GD [5] and SpecGD in the setting considered in the previous section. For tractability, we make two simplifications: we consider squared-loss

objective instead of cross-entropy loss, and population setting instead of finite samples. Specifically, we let total loss $\mathcal{L}(\boldsymbol{W}) = \frac{1}{2}\mathbb{E}\|\boldsymbol{y} - \boldsymbol{W}\boldsymbol{x}\|^2$, where the expectation is over the joint distribution of $\boldsymbol{x}, y$ in (say) (DM). In addition to this, define $\mathcal{L}_c(\boldsymbol{W}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}|y=c}\|\boldsymbol{y} - \boldsymbol{W}\boldsymbol{x}\|^2$ to be the class-conditional loss for class $c \in [k]$ and let $\mathcal{L}_{\text{bal}}(\boldsymbol{W}) = \frac{1}{k}\sum_{c\in[k]}\mathcal{L}_c(\boldsymbol{W})$ be the balanced loss. Define the population moment matrices $\boldsymbol{\Sigma_x} := \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\top\right]$ and $\boldsymbol{\Sigma_{xy}} := \mathbb{E}\left[\boldsymbol{y}\,\boldsymbol{x}^\top\right]$. We focus on a setting where these satisfy the following assumption: the (full) SVDs of these moment matrices are jointly diagonalizable.

Assumption 1 *There exist orthonormal matrices $\boldsymbol{U} \in \mathbb{R}^{k\times k}, \boldsymbol{V} \in \mathbb{R}^{d\times d}$ and matrices $\boldsymbol{S} \in \mathbb{R}^{k\times d}, \boldsymbol{\Lambda} \in \mathbb{R}^{d\times d}$ with non-zero entries only along their main diagonals ($\sigma_1 \geq \sigma_2 \cdots \geq \sigma_k \geq 0$ and $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_d \geq 0$, respectively), such that, $\boldsymbol{\Sigma_{xy}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ and $\boldsymbol{\Sigma_x} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$.*

This assumption is adopted by Gidel et al. [7], Saxe et al. [17], who apply it to empirical moment matrices to derive closed-form training dynamics of two-layer linear networks. Here, we instead apply this assumption on population data and analyze the population loss making it possible to study test statistics. In the lemma below we show that this assumption holds under data following (DM).

Lemma 1 *The population moment matrices of data model (DM) satisfy Assumption 1.*

Evolution of $\boldsymbol{W}_t$. We compare the evolution of the weight matrix $\boldsymbol{W}_t$ for GD and SpecGD over iterations t. For each iteration t, define $\overline{\boldsymbol{W}}_t := \boldsymbol{U}^\top\boldsymbol{W}_t\boldsymbol{V}$ and recall that $\overline{W}_t[i,i]$ denotes the i-th main-diagonal entry of $\overline{\boldsymbol{W}}_t$, for $i \in [k]$.

For GD, under Assumption 1 it is shown in Saxe et al. [17] (see also Appendix B.2) that when initialized at zero and run with sufficiently small step size, its iterates $\boldsymbol{W}_t$ are such that $\overline{\boldsymbol{W}}_t$ is diagonal at each iteration with diagonal entries evolving as (with the approximation becoming accurate as $\eta \to 0$):

$$\overline{W}_t[i,i] \approx \tfrac{\sigma_i}{\lambda_i}\left(1 - e^{-\eta\lambda_i t}\right).$$

The following result characterizes the dynamics of $\boldsymbol{W}_t$ for SpecGD.

Proposition 2 *Assume zero initialization $\boldsymbol{W_0} = \boldsymbol{0}$ and Assumption 1 holds. Then, for SpecGD, at each iteration, $\boldsymbol{W}_t = \boldsymbol{U}\overline{\boldsymbol{W}}_t\boldsymbol{V}^\top$ where $\overline{\boldsymbol{W}}_t$ is zero except its main diagonal along which entries evolve as follows for $i \in [k]$:*

$$\overline{W}_t[i,i] = \eta\, t\, \mathbb{1}\left[t \leq \tfrac{\sigma_i}{\eta\lambda_i}\right] + \tfrac{\sigma_i}{\lambda_i}\,\mathbb{1}\left[t > \tfrac{\sigma_i}{\eta\lambda_i}\right].$$

Comparing the above two displays, which contrast GD's and SpecGD's iterate evolution, shows the following. Although both methods asymptotically converge to the same solution, their trajectories differ significantly. While GD learns component i at a rate proportional to λ_i, meaning more dominant components are learned faster. In contrast, SpecGD learns all components at the same rate until each individual value saturates and converges to its terminal value.

We now show that this property of SpecGD to learn concepts at equal rate translates to superior generalization in an imbalanced setting where least-significant spectral components of the moment matrices correspond to learning minority features. Concretely, under data model (DM), the k first eigenvectors of $\boldsymbol{\Sigma}_x$ align with the class-mean vectors, ordered in decreasing class prior probability (see proof of Lemma 1). Intuitively, learning least-significant components earlier during training should translate to generalization gains. The following theorem formalizes this intuition.

Theorem 3 *Assume data model (DM), zero initialization, and equal sufficiently small step size η for GD and SpecGD. Let $t^\star = \frac{\sigma_m}{\eta\lambda_m}$ be the first time SpecGD fits a minority component of class-prior p_m. Further assume $\sigma_x^2 \in \left[7p_m, \frac{1-p_m}{k}\right]$. Then for every $t \in (0, t^\star]$, SpecGD dominates GD with a growing loss gap:*

$$\mathcal{L}_m^{\text{GD}}(t) - \mathcal{L}_m^{\text{Spec}}(t) \geq \tfrac{5\eta(1-p_m)}{4}t.$$

Moreover, in a STEP-imbalance setting with $k_M \leq \frac{(1-p_m)}{4}k$ majority classes of prior p_M and $k_m = k - k_M$ minority classes of prior p_m, the gap in the balanced loss satisfies

$$\mathcal{L}_{\text{bal}}^{\text{GD}}(t) - \mathcal{L}_{\text{bal}}^{\text{Spec}}(t) \geq \tfrac{\eta(1-p_m)}{4}t.$$

4. Future Work

There are many exciting directions for future work. First, we aim to extend our analysis to the group-imbalanced spurious correlations setting of Sec. 2. Second, even in the linear setting of Sec. 3, proving the empirically observed superior performance of SpecGD over NGD and SignGD remains open. Third, while the minimal setting studied here already provides new insights into SpecGD's bias toward learning principal directions at equal rates, we wish to investigate how these benefits manifest in more practical large-scale settings.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020.

[2] Dennis S Bernstein. *Matrix mathematics: theory, facts, and formulas*. Princeton university press, 2009.

[3] Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[5] Augustin-Louis Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus de l'Académie des Sciences de Paris*, 25:536–538, 1847.

[6] Chen Fan, Mark Schmidt, and Christos Thrampoulidis. Implicit bias of signgd and adam on multiclass separable data. *arXiv preprint arXiv:2502.04664*, 2025.

[7] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[8] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2018.

[9] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

[10] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Frederik Kunstner, Robin Yadav, Alan Milligan, Mark Schmidt, and Alberto Bietti. Heavy-tailed class imbalance and why adam outperforms gradient descent on language models. *arXiv preprint arXiv:2402.19449*, 2024.

[13] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

[14] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. 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=aExAsh1UHZo.

[15] Harold Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586.

[16] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[17] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[18] Hao-Jun Michael Shi, Tsung-Hsien Lee, Shintaro Iwasaki, Jose Gallego-Posada, Zhijing Li, Kaushik Rangadurai, Dheevatsa Mudigere, and Michael Rabbat. A distributed data-parallel pytorch implementation of the distributed shampoo optimizer for training neural networks at-scale. https://github.com/facebookresearch/optimizers/tree/main/distributed_shampoo, 2023.

[19] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19 (70):1–57, 2018.

[20] Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: Improving and stabilizing shampoo using adam for language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IDxZhXrpNf.

[21] Chenyang Zhang, Difan Zou, and Yuan Cao. The implicit bias of adam on separable data. *arXiv preprint arXiv:2406.10650*, 2024.

Appendix A. Optimizers

In this section, we list the update rules for all the optimizers considered in the paper, for completeness. We start with the update rules of NGD, SpecGD and SignGD, respectively, and then write the updates for Shampoo and Adam.

Using the notation introduced in Section 3, we have

NGD updates: $\boldsymbol{\Delta}_t = \frac{\boldsymbol{\nabla}_t}{\|\boldsymbol{\nabla}_t\|_F}$.

Let the truncated SVD of $\boldsymbol{\nabla}_t$ be written as $\boldsymbol{U}_t\boldsymbol{\Sigma}_t\boldsymbol{V}_t^\top$, where $\boldsymbol{U}_t$ and $\boldsymbol{V}_t$ are orthonormal matrices and $\boldsymbol{\Sigma}_t$ is a positive diagonal matrix.

SpecGD updates: $\boldsymbol{\Delta}_t = \boldsymbol{U}_t\boldsymbol{V}_t^\top$.

SignGD updates: $\boldsymbol{\Delta}_t = \mathtt{sign}(\boldsymbol{\nabla}_t)$, where $\mathtt{sign}(x) := \frac{x}{|x|}$ and $\mathtt{sign}(()0) = 0$, and it is applied element-wise on the matrix $\boldsymbol{\nabla}_t$.

For Shampoo, first define the preconditioning matrices

$$\boldsymbol{L}_t = \beta_2\boldsymbol{L}_{t-1} + (1-\beta_2)\boldsymbol{\nabla}_t\boldsymbol{\nabla}_t^\top \quad \text{and} \quad \boldsymbol{R}_t = \beta_2\boldsymbol{R}_{t-1} + (1-\beta_2)\boldsymbol{\nabla}_t^\top\boldsymbol{\nabla}_t,$$

where the parameter β_2 denotes the preconditioning accumulation parameter. These preconditioners are used to give the following update.

Shampoo updates: $\boldsymbol{\Delta}_t = \boldsymbol{L}_t^{-1/4}\boldsymbol{\nabla}_t\boldsymbol{R}_t^{-1/4}$. It is easy to see that Shampoo reduces to SpecGD when we set $\beta_2 = 0$.

For Adam, let $\hat{\boldsymbol{M}}_t = \frac{\boldsymbol{M}_{t+1}}{1-\beta_1^{t+1}} = \frac{1}{1-\beta_1^{t+1}}\Big(\beta_1\boldsymbol{M}_t + (1-\beta_1)\boldsymbol{\nabla}_t\Big)$ denote the bias-corrected first-moment estimate, and $\hat{\boldsymbol{Z}}_t = \frac{\boldsymbol{Z}_{t+1}}{1-\beta_2^{t+1}} = \frac{1}{1-\beta_2^{t+1}}\Big(\beta_2\boldsymbol{Z}_t + (1-\beta_2)\boldsymbol{\nabla}_t \odot \boldsymbol{\nabla}_t\Big)$ denote the bias-corrected second (raw) moment estimate, where $\odot$ denotes the Hadamard product, and β_1, β_2 denote the momentum parameters.

Adam updates: $\boldsymbol{\Delta}_t = \dfrac{\hat{\boldsymbol{M}}_t}{\hat{\boldsymbol{Z}}_t + \epsilon\mathbf{1}\mathbf{1}^\top}$, where the division is done element-wise, $\epsilon > 0$ is the numerical precision parameter, and $\mathbf{1}$ denotes the all-ones vector. It is easy to see that Adam reduces to SignGD when we set $\beta_1 = \beta_2 = \epsilon = 0$.

Appendix B. Proofs

B.1. Proof of Lemma 1

The covariance matrix is

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{x}} \;&:=\; \mathbb{E}\big[\boldsymbol{x}\boldsymbol{x}^\top\big] = \mathbb{E}_{y,\boldsymbol{\varepsilon}}\Big[(\boldsymbol{\mu}_y+\boldsymbol{\varepsilon})(\boldsymbol{\mu}_y+\boldsymbol{\varepsilon})^\top\Big] \\ &= \underbrace{\mathbb{E}_y\big[\boldsymbol{\mu}_y\boldsymbol{\mu}_y^\top\big]}_{=:\boldsymbol{\Sigma}_{\boldsymbol{\mu}}} + \mathbb{E}_y\big[\boldsymbol{\mu}_y\big]\mathbb{E}_{\boldsymbol{\varepsilon}}\big[\boldsymbol{\varepsilon}^\top\big] \;+\; \mathbb{E}_{\boldsymbol{\varepsilon}}\big[\boldsymbol{\varepsilon}\big]\mathbb{E}_y\big[\boldsymbol{\mu}_y^\top\big] \;+\; \mathbb{E}\big[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top\big] = \boldsymbol{\Sigma}_\mu \;+\; \sigma_x^2\boldsymbol{I}_d,\end{aligned}$$

since $\boldsymbol{\varepsilon}$ and y are independent and $\mathbb{E}[\boldsymbol{\varepsilon}] = 0$. Further,

$$\boldsymbol{\Sigma}_{\boldsymbol{\mu}} \;=\; \sum_{c=1}^{k} p_c\,\boldsymbol{\mu}_c\boldsymbol{\mu}_c^\top \;=\; \boldsymbol{M}\boldsymbol{P}\boldsymbol{M}^\top, \qquad \boldsymbol{M} := \big[\,\boldsymbol{\mu}_1\;\boldsymbol{\mu}_2\;\cdots\;\boldsymbol{\mu}_k\big] \in \mathbb{R}^{d\times K},\; \boldsymbol{P} := \mathrm{diag}(p_1,\ldots,p_k).$$

Then, we can write $\boldsymbol{\Sigma_x} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$, where

$$\boldsymbol{\Lambda} = \mathrm{diag}(p_1 + \sigma_x^2, \dots, p_k + \sigma_x^2, \underbrace{\sigma_x^2, \dots, \sigma_x^2}_{d-k \text{ times}}), \tag{2}$$
$$\boldsymbol{V} = \big[\, \boldsymbol{M}\ \boldsymbol{V}_\perp \big] = \big[\, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k,\ \boldsymbol{v}_{k+1}, \dots, \boldsymbol{v}_d \big] \in \mathbb{R}^{d\times d}, \quad \{\boldsymbol{v}_i\}_{i=k+1}^d \perp \{\boldsymbol{\mu}_c\}_{c=1}^k.$$

Here, we used the assumption on orthonormality of the means.

The cross-covariance is

$$\boldsymbol{\Sigma_{xy}} := \mathbb{E}\big[\boldsymbol{y}\,\boldsymbol{x}^\top\big] = \sum_{c=1}^k p_c\,\boldsymbol{y}_c\,\mathbb{E}\big[\boldsymbol{x}^\top \mid y = c\big] = \sum_{c=1}^k p_c\,\boldsymbol{e}_c\,\boldsymbol{\mu}_c^\top.$$

We can write $\boldsymbol{\Sigma_{xy}} = \boldsymbol{U}\,\boldsymbol{S}\,\boldsymbol{V}^\top$, where

$$\boldsymbol{U} = \boldsymbol{I}_k, \quad \boldsymbol{S} = \mathrm{diag}(p_1, \dots, p_k, 0, \dots, 0). \tag{3}$$

B.2. Proof of Prop. 2

We can write the gradient as

$$\nabla\mathcal{L}(\boldsymbol{W}_t) = -\mathbb{E}\left[\big(\boldsymbol{y} - \boldsymbol{W}_t\boldsymbol{x}\big)\boldsymbol{x}^\top\right] = -\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top + \boldsymbol{W}_t\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top.$$

For GD, we have

$$\begin{aligned}
\boldsymbol{W}_{t+1} &= \boldsymbol{W}_t - \eta\nabla\mathcal{L}(\boldsymbol{W}_t) \\
&= \boldsymbol{W}_t + \eta\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top - \eta\boldsymbol{W}_t\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top = \boldsymbol{W}_t\big(\boldsymbol{I} - \eta\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top\big) + \eta\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top \\
&= \boldsymbol{W}_0\big(\boldsymbol{I} - \eta\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top\big)^{t+1} + \sum_{\tau=0}^t \eta\boldsymbol{U}\boldsymbol{S}\big(\boldsymbol{I} - \eta\boldsymbol{\Lambda}\big)^\tau\boldsymbol{V}^\top.
\end{aligned}$$

Since $\boldsymbol{W}_0 = \boldsymbol{0}$, this gives

$$\overline{\boldsymbol{W}}_{t+1} := \boldsymbol{U}^\top\boldsymbol{W}_{t+1}\boldsymbol{V} = \eta\boldsymbol{S}\sum_{\tau=0}^t\big(\boldsymbol{I} - \eta\boldsymbol{\Lambda}\big)^\tau.$$

Assuming $\eta < \frac{1}{d_{\max}}$, we get

$$\overline{W}_{t+1}[i,i] = \eta\sigma_i\sum_{\tau=0}^t\big(1 - \eta\,\lambda_i\big)^\tau = \sigma_i\,\frac{1 - \big(1 - \eta\lambda_i\big)^{\,t+1}}{\lambda_i}.$$

For sufficiently small η this gives the approximation

$$\overline{W}_{t+1}[i,i] \approx \frac{\sigma_i}{\lambda_i}\big(1 - e^{-\eta\lambda_i(t+1)}\big).$$

For SpecGD, note that the gradient can be written in terms of $\overline{\boldsymbol{W}}_t$ as

$$\nabla_t = \nabla\mathcal{L}(\boldsymbol{W}_t) = -\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top + \boldsymbol{W}_t\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top = \boldsymbol{U}(\boldsymbol{\Lambda}\overline{\boldsymbol{W}}_t - \boldsymbol{S})\boldsymbol{V}^\top.$$

Starting at $\boldsymbol{W}_t = 0 \Leftrightarrow \overline{\boldsymbol{W}}_t = 0$ gives $\nabla_0 = -\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$.

Thus, $\boldsymbol{W}_1 = \eta \boldsymbol{U}\boldsymbol{V}^\top \Rightarrow \overline{\boldsymbol{W}}_1[i,j] = \eta \begin{cases} 1 & i = j \in [k] \\ 0 & i \neq j \end{cases}$. Proceeding this way, we arrive at the following update rule for all t,

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \eta \sum_{i:\overline{W}_t[i,i]\,\lambda_i < \sigma_i} \boldsymbol{u}_i \boldsymbol{v}_i^\top.$$

This gives

$$\overline{W}_{t+1} = \overline{W}_t + \eta \sum_{i:\,\overline{W}_t[i,i]\,\lambda_i < \sigma_i} \boldsymbol{e}_i \boldsymbol{e}_i^\top.$$

Since $\boldsymbol{W}_0 = \boldsymbol{0}$, we concled with the desired:

$$\overline{W}_{t+1}[i,i] = \eta\,(t+1)\,\mathbb{1}\left[(t+1) \leq \tfrac{\sigma_i}{\eta\lambda_i}\right] + \tfrac{\sigma_i}{\lambda_i}\,\mathbb{1}\left[(t+1) > \tfrac{\sigma_i}{\eta\lambda_i}\right].$$

B.3. Proof of Theorem 3

The population loss for class c using iterate $\boldsymbol{W}_t$ is written as

$$\begin{aligned} \mathcal{L}_c(t) :=\ & \mathbb{E}\big\|\boldsymbol{y} - \boldsymbol{W}_t\,\boldsymbol{x}\big\|_2^2 = \mathbb{E}\big[\,1 - 2\boldsymbol{y}^\top \boldsymbol{W}_t\,\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{W}_t^\top \boldsymbol{W}_t\,\boldsymbol{x}\big]. \\ &= 1 - 2\,\boldsymbol{e}_c^\top \boldsymbol{W}_t\,\boldsymbol{\mu}_c + \big\|\boldsymbol{W}_t\,\boldsymbol{\mu}_c\big\|_2^2 + \sigma_x^2\,\|\boldsymbol{W}_t\|_F^2 \\ &= \big\|\boldsymbol{e}_c - \boldsymbol{W}_t\,\boldsymbol{\mu}_c\big\|_2^2 + \sigma_x^2\,\|\boldsymbol{W}_t\|_F^2. \end{aligned}$$

Using Prop. 2 and Lem. 1, shows that the singular values of $\boldsymbol{W}_t^{\text{GD}}$ and $\boldsymbol{W}_t^{\text{SpecGD}}$ evolve as

$$\sigma_c^{\text{GD}}(t) := \overline{W}_t^{\text{GD}}[c,c] = \alpha_c\left(1 - \exp\left(-\tfrac{\eta\,p_c}{\alpha_c}t\right)\right), \tag{4}$$

$$\sigma_c^{\text{Spec}}(t) := \overline{W}_t^{\text{SpecGD}}[c,c] = \eta t \mathbb{1}\left[t \leq \tfrac{\alpha_c}{\eta}\right] + \alpha_c \mathbb{1}\left[t > \tfrac{\alpha_c}{\eta}\right], \tag{5}$$

where $\alpha_c := \frac{\sigma_c}{\lambda_c}$ denotes the ratio of the singular values for class c, and using Eqs. (2) and (3) from the proof of Lem. 1, $\alpha_c = \frac{p_c}{\sigma_x^2 + p_c}$.

Using these expressions, we can write the per-class loss in terms of the singular values of $\boldsymbol{W}_t$ as

$$\mathcal{L}_c(t) = (1 - \sigma_c(t))^2 + \sigma_x^2 \sum_{c=1}^{K} \sigma_c^2(t). \tag{6}$$

The time derivatives of $\sigma_c^{\text{GD}}(t)$ and $\sigma_c^{\text{Spec}}(t)$ for $t \leq t^*$ are given by

$$s_c^{\text{GD}}(t) := \tfrac{d\sigma_c^{\text{GD}}(t)}{dt} = p_c \eta\, e^{-(\sigma_x^2 + p_c)\eta t}, \qquad s_c^{\text{Spec}}(t) := \tfrac{d\sigma_c^{\text{Spec}}(t)}{dt} = \eta.$$

Define the gaps

$$\Delta\sigma_c(t) := \sigma_c^{\text{Spec}}(t) - \sigma_c^{\text{GD}}(t) \geq 0, \qquad \Delta s_c(t) := s_c^{\text{Spec}}(t) - s_c^{\text{GD}}(t) = \eta - \eta\,p_c\,e^{-(\sigma_x^2 + p_c)\eta t} > 0. \tag{7}$$

Also note that $\Delta s_c(t)$ is *increasing* in t.

Minority–class loss gap. For a fixed t, consider the minority loss gap

$$\Delta\mathcal{L}_m(t) := \mathcal{L}_m^{\text{GD}}(t) - \mathcal{L}_m^{\text{Spec}}(t). \tag{8}$$

Using the per-class-loss from Eq. (6) and differentiating,

$$\Delta\mathcal{L}'_m(t) = -2\underbrace{\left[(1-\sigma_m^{\text{GD}})\, s_m^{\text{GD}} - (1-\sigma_m^{\text{Spec}})\, s_m^{\text{Spec}}\right]}_{\text{Term-1}(\Phi)} + 2\sigma_x^2 \underbrace{\left[\sum_j \sigma_j^{\text{GD}} s_j^{\text{GD}} - \sum_j \sigma_j^{\text{Spec}} s_j^{\text{Spec}}\right]}_{\text{Term-2}(\Psi)}, \tag{9}$$

where we drop the time t in $\sigma(t)$ and $s(t)$ for brevity.

Term-1 (Φ). Add–subtract $(1-\sigma_m^{\text{GD}})s_m^{\text{Spec}}$ to Term-1, and we have

$$\Phi = (1-\sigma_m^{\text{GD}})s_m^{\text{GD}} - (1-\sigma_m^{\text{Spec}})s_m^{\text{Spec}} = -(1-\sigma_m^{\text{GD}})\,\Delta s_m + \Delta\sigma_m\, s_m^{\text{Spec}}. \tag{10}$$

We know that

$$\Delta\sigma(t) = \int_0^t \Delta s(\tau)\, d\tau \le t\,\Delta s(t) \quad \text{(increasing integrand)}.$$

Substituting in Eq. (10), we get

$$\begin{aligned}\Phi &\le -\big(1-\sigma_m^{\text{GD}} - t\, s_m^{\text{Spec}}\big)\Delta s_m(t)\\ &\le -\big(1-\alpha_m-\eta t\big)\Delta s_m(t)\\ &\le -\big(1-2\alpha_m\big)\Delta s_m(t).\end{aligned} \tag{11}$$

Term-2 (Ψ) Before any σ_j^{Spec} saturates,

$$\sum_j \sigma_j^{\text{Spec}} s_j^{\text{Spec}} = k\eta^2 t,$$

which gives for all $t \le t^*$ that

$$\Psi \ge -\eta k \alpha_m. \tag{12}$$

Using Eq. (11) and Eq. (12) in Eq. (9), we have

$$\begin{aligned}\Delta\mathcal{L}'_m(t) &\ge 2(1-2\alpha_m)\Delta s_m(t) - 2\sigma_x^2\eta k\alpha_m\\ &\ge 2(1-2\alpha_m)\,\eta(1-p_m) - 2\sigma_x^2\eta k\alpha_m\\ &\ge \tfrac{5\eta(1-p_m)}{4},\end{aligned} \tag{13}$$

since $\alpha_m \le \frac{1}{8}$ (as $p_m \le \frac{\sigma_x^2}{7}$), and $\sigma_x^2 \le \frac{1-p_m}{k}$. Since $\Delta\mathcal{L}_m(0) = 0$, integrating over $(0, t]$ gives the final bound.

Class–balanced loss gap. For a fixed t, the class-balanced loss gap is

$$\Delta\mathcal{L}_{\text{bal}}(t) := \sum_c \mathcal{L}_c^{\text{GD}}(t) - \mathcal{L}_c^{\text{Spec}}(t). \tag{14}$$

Here, we consider k_M majority classes with $p_c = p_{\text{M}}$ and $k_m = k - k_M$ minority classes with $p_c = p_{\text{m}}$. Using the per-class-loss from Eq. (6) and differentiating,

$$\Delta\mathcal{L}'_{\text{bal}}(t) = -2\tfrac{k_M}{k}\underbrace{\left[(1-\sigma_M^{\text{GD}})\, s_M^{\text{GD}} \;-\; (1-\sigma_M^{\text{Spec}})\, s_M^{\text{Spec}}\right]}_{\text{Term-3}(\Lambda)} \;-\; 2\tfrac{k_m}{k}\Phi \;+\; 2\sigma_x^2\,\Psi, \tag{15}$$

Term-3 (Λ). Following similar steps as we used for Term-1 (Φ), we have

$$\begin{aligned}\Lambda \;&=\; -(1-\sigma_M^{\text{GD}})\,\Delta s_M + \Delta\sigma_M\, s_M^{\text{Spec}} \\ &\leq\; -\big(1-2\alpha_M\big)\Delta s_M(t) \leq \eta(1-p_M e^{-\frac{\alpha_m}{\alpha_M}p_M}) \leq \eta.\end{aligned} \tag{16}$$

Also we know,

$$-\Phi \geq (1-2\alpha_m)\Delta s_m(t) \;\geq\; -\eta(1-p_m e^{-p_m}) \geq \eta. \tag{17}$$

Using Eq. (11), Eq. (12), Eq. (16) and Eq. (17) in Eq. (15), we have

$$\begin{aligned}\Delta\mathcal{L}'_{\text{bal}}(t) &\geq 1.25\eta(1-p_m) - 4\eta\tfrac{k_M}{k}, \\ &\geq \tfrac{\eta(1-p_m)}{4},\end{aligned} \tag{18}$$

since $\frac{k_M}{k} \leq \frac{1-p_m}{4}$. Since $\Delta\mathcal{L}_{\text{bal}}(0) = 0$, integrating over $(0, t]$ gives the final bound.

Appendix C. Results on MNIST with Heavy-tailed Class Imbalance

In this section, we consider the Barcoded MNIST dataset, which is a variant of MNIST with heavy-tailed class imbalance introduced in Kunstner et al. [12]. Barcoded MNIST contains two types of classes: 10 classes with 5000 samples each from the original MNIST dataset (majority), and 10×2^{10} additional classes with 5 samples each (minority). The images in the minority classes are generated by taking images from the original MNIST dataset, and encoding a 10-bit barcode into the top left corner of the images, for each of the 10 original classes.

Following Kunstner et al. [12], we train a 2-layer CNN on this dataset in the full-batch setting. In Fig. 3, we compare the total train loss as well as train loss on the majority and minority classes (each comprising about $\approx 50\%$ of the total samples) for the three optimizers: GD with momentum, Shampoo and Adam (we use `learning rates` 0.005, 10^{-4} and 10^{-4}, respectively).

We rely on the `distributed-shampoo` reference implementation of Shi et al. [18] and use default β parameters $(\beta_1, \beta_2) = (0.9, 1.0)$, matrix inverse stabilization parameter $\varepsilon = 10^{-8}$, `precondition_frequency`$= 1$, `max_preconditioner_dim`$= 8192$. In order to stabilize training, we also use `AdamGraftingConfig` for the update grafting. Here, β_1 governs the exponential moving average of the raw gradients, while β_2 governs the accumulation in the matrix preconditioners (see Appendix A). Further, ε is a small diagonal 'jitter' that stabilises the matrix inverse, and Adam grafting simply rescales the Shampoo update so its overall magnitude matches that of a plain Adam step (with default β parameters).

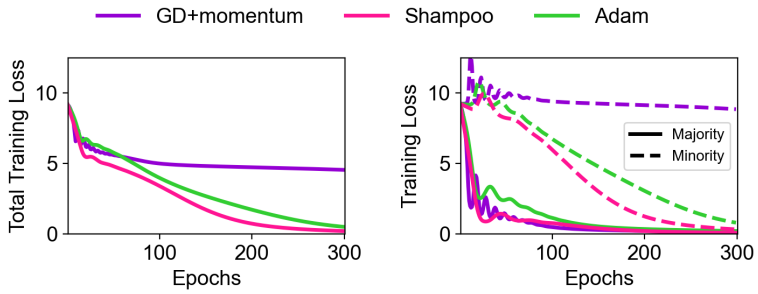

Figure 3: Comparison of GD, Shampoo and Adam when training a CNN on the Barcoded MNIST dataset from Kunstner et al. [12], a variant of MNIST with heavy-tailed class imbalance. GD only drives the loss on majority classes towards 0 and makes little progress on the minority classes. In contrast, Shampoo and Adam drive the loss on both majority and minority classes towards 0.

Consistent with the results in Kunstner et al. [12], we observe while that GD only minimizes the loss on the majority classes, and makes negligible progress on the minority classes, Adam minimizes loss on both majority and minority classes. In addition, we find that training with Shampoo has a similar behaviour as Adam: it also minimizes the loss on both majority and minority classes, in contrast to GD.

Appendix D. Details of Experimental Settings

Coloured-MNIST spurious-correlation experiment. We train a two-layer multilayer perceptron (MLP) with hidden width $m = 128$ and ReLU activation, using inputs from the standard Coloured-MNIST dataset. Each image is 28×28 with 3 channels (RGB), The final output head is a linear layer initialised to $\pm 1/\sqrt{m}$ and kept frozen throughout training to isolate representation learning in the hidden layer.

We compare three optimisers: Shampoo (`lr`$= 0.1$), SGD (`lr`$= 0.1$, momentum $\beta = 0.9$), and Adam (`lr`$= 0.01$, default β parameters). Adam is the only optimizer for which cosine learning rate decay is applied; the others use a constant schedule. Training is performed for 500 `epochs` with a `batch size` of 64.

All Shampoo hyper-parameters mirror those in Appendix C, except that we do the preconditioning every five steps (`precondition_frequency`$= 5$), set the stabilisation term to $\varepsilon = 10^{-6}$, and replace Adam grafting with `SGDGraftingConfig`. For each run, we log training loss and accuracy, along with test accuracy on both the minority (colour-flipped) and majority (colour-aligned) groups.