Why Loss Re-weighting Works If You Stop Early: Training Dynamics of Unconstrained Features

Yize Zhao [1] **Christos Thrampoulidis** [1]

Abstract

The application of loss reweighting in modern deep learning presents a nuanced picture. While it fails to alter the terminal learning phase in overparameterized deep neural networks (DNNs) trained on high-dimensional datasets, empirical evidence consistently shows it offers significant benefits early in training. To transparently demonstrate and analyze this phenomenon, we introduce a small-scale model (SSM). This model is specifically designed to abstract the inherent complexities of both the DNN architecture and the input data, while maintaining key information about the structure of imbalance within its spectral components. On the one hand, the SSM reveals how vanilla empirical risk minimization preferentially learns to distinguish majority classes over minorities early in training, consequently delaying minority learning. In stark contrast, reweighting restores balanced learning dynamics, enabling the simultaneous learning of features associated with both majorities and minorities.

1. Introduction

Classification of imbalanced datasets is a pervasive challenge in practical machine learning. Most real-world datasets exhibit classes with varying numbers of examples, where classes with significantly fewer training examples are commonly referred to as minorities. A classical and widely adopted approach to learning from imbalanced data is *reweighting*, which modifies the loss function to effectively increase the contribution of minority class examples during training. Concretely, instead of a learning model f that minimizes vanilla empirical risk $\mathcal{L}(f) = \frac{1}{n}\sum_{i\in[n]} \ell\left(f(\boldsymbol{x}_i), y_i\right)$, over training examples of feature-label pairs $(\boldsymbol{x}_i, y_i)$, loss reweighting minimizes

$$\mathcal{L}_{\mathrm{RW}}(f) = \frac{1}{n}\sum\nolimits_{i\in[n]} \omega_{y_i} \cdot \ell(f(\boldsymbol{x}_i), y_i), \tag{1}$$

where $\omega_{y_i} \propto 1/\hat{\pi}_{y_i}$ is typically set proportional to the inverse empirical frequency[1] of the class $y_i \in [k]$ to which example $i \in [n]$ belongs. This approach has historically been popular not only due to its simplicity but also because of its statistical optimality: in the population limit, $\mathcal{L}_{\mathrm{RW}}$ is known to optimize for balanced accuracy, a metric that weighs class-conditional accuracies equally rather than by their class frequencies (see Appendix A.1 for further background).

However, modern machine learning practice, particularly with large-scale deep neural networks (DNNs) and complex high-dimensional datasets, presents a more nuanced picture regarding the statistical optimality and practical efficacy of $\mathcal{L}_{\mathrm{RW}}$. On one hand, empirical observations (Byrd and Lipton, 2019) and subsequent theoretical justifications (Sagawa et al., 2020; Kini et al., 2021; Xu et al., 2021) have demonstrated that $\mathcal{L}_{\mathrm{RW}}$ often fails to substantially improve the accuracy of minority classes over vanilla $\mathcal{L}$ if model training continues for a large number of iterations until convergence (further details can be found in Appendix A.2). On the other hand, empirical evidence also suggests that $\mathcal{L}_{\mathrm{RW}}$ can improve performance over vanilla $\mathcal{L}$ when model training is subjected to early stopping (Byrd and Lipton, 2019; Xu et al., 2021) Complementary to this, emphasizing the early-training benefits of reweighting, it has also been found that it can further boost the performance of alternative modern loss functions specifically proposed for imbalanced classification (Cao et al., 2019; Menon et al., 2020; Kini et al., 2021; Ye et al., 2020; Li et al., 2021).

[1]Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada. Correspondence to: Christos Thrampoulidis <cthrampo@ece.ubc.ca>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

[1]In practice, it is common setting $\omega_{y_i} \propto 1/\hat{\pi}_{y_i}^{\gamma}$ where $\gamma \in (0, 1)$ is tuned empirically, e.g., (Cao et al., 2019; Menon et al., 2020).

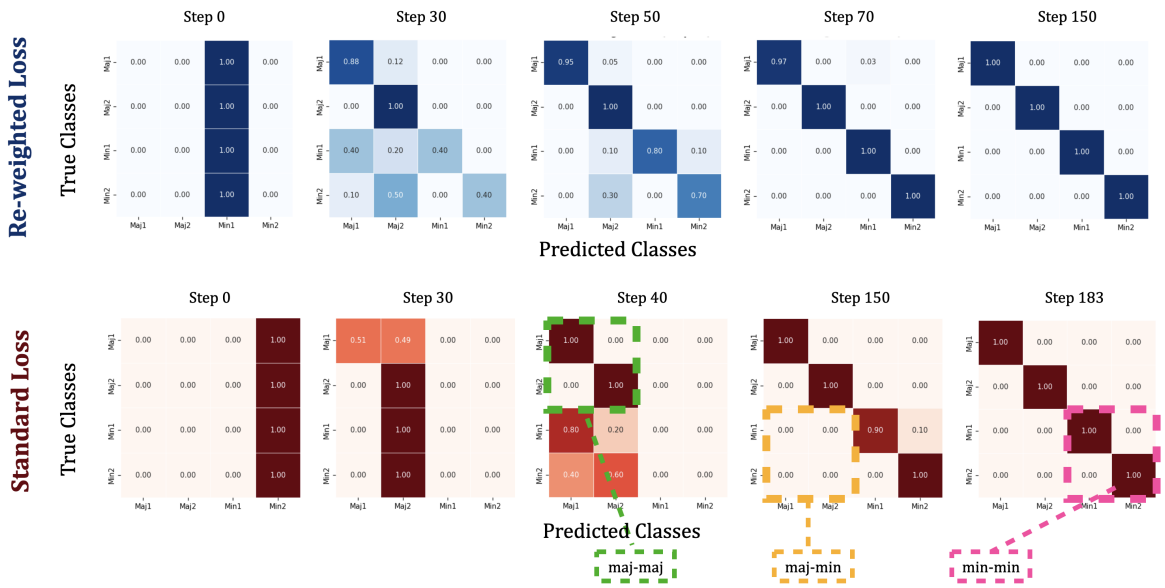

Figure 1. **Confusion matrix evolution during training on imbalanced MNIST** (2 majorities, 2 minorities, imbalance ratio $R = 10$). Top row: model trained with **reweighted loss**. Bottom row: model trained with **standard loss**. Under **standard loss** (bottom), the model exhibits staged learning dynamics: it first learns to distinguish between majority classes (green boxes) by step 40, then separates majority from minority classes (orange boxes) by step 150, and only later distinguishes between minority classes (pink boxes) at step 183. This progression reflects the ordering of singular values in the simplex-encoded label (SEL) matrix. In contrast, under **reweighted loss** (top), the model learns class distinctions more uniformly and earlier, with balanced improvements across majorities and minorities by step 70. Our small-scale model replicates this behavior and attributes it to reweighting effectively flattens the spectrum of the labels.

To the best of our knowledge, a clear understanding of these early-training benefits of $\mathcal{L}_{\mathrm{RW}}$ is still lacking:

Why does reweighting help early in training? How does it modify the learning dynamics to favor minorities?

The inherent complexity of DNNs and of real-world datasets makes it challenging to directly answer these questions.

Our contribution is to identify and analyze a small-scale model (SSM) that transparently demonstrates the exact impact of reweighting on training dynamics, specifically favoring minority classes.

In Section 2, as a motivating experiment, we compare how confusion matrices evolve when training without and with reweighting on a real dataset. This reveals the strong and rapid impact of reweighting at early training stages; see Fig. 1. In Section 3, we introduce our SSM with the explicit goal of explaining this empirically observed behavior. With necessary abstractions of the data, the DNN, and the loss function, and appropriate interpretations of its spectral components, our SSM effectively replicates the empirically observed DNN behaviors related to reweighting. Finally, Section 4 leverages the inherent simplicity of the SSM to yield a theoretical understanding of these phenomena.

2. Motivating Experiment

We empirically illustrate the different learning dynamics induced by vanilla empirical risk minimization (ERM) and loss reweighting in an imbalanced classification setting. We construct a 4-class subset of the MNIST dataset with synthetic imbalance: two classes are designated as majorities with 100 training samples each, and the remaining two as minorities with 10 samples each (imbalance ratio $R = 10$). We train a 3-layer convolutional neural network (CNN) with embedding dimension $d = 32$ under both standard cross-entropy (CE) and reweighted CE, using the Adam optimizer with learning rate 10^{-3} and batch size 64. For reweighting, class weights are set inversely proportional to class frequencies.

Figure 1 vividly illustrates the early benefits of loss reweighting. By tracking the confusion matrix on the training data across iterations, we observe the following distinct learning behaviors:

1. **Vanilla ERM exhibits a clear preferential learning order favoring majorities**: The model first learns to correctly classify majority classes, then gradually learns to distinguish minority classes from majority classes, and only in later stages begins to differentiate between various minority classes themselves.
2. **Reweighting eliminates this preferential ordering, enabling earlier classification of minorities.**

See Appendix B for additional experiments, including the evolution of other training and test metrics (e.g., test-time confusion matrices and balanced accuracy) during training.

3. Small-Scale Model

3.1. Model Description

Our model abstracts the inherent complexities of both the DNN architecture and the input data by using a model of unconstrained features (UFM). It further combines this with a simplification on the loss from cross-entropy to squared-loss. For concreteness, we further focus on a STEP-imbalanced setting.

Architecture/data abstraction. The UFM substitutes the training embeddings $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$, which are typically high-dimensional representations generated by the last activation layer of a DNN parameterized by $\boldsymbol{\theta}$, with directly trainable vectors $\boldsymbol{h}_i \in \mathbb{R}^d$. This abstraction simultaneously bypasses the network architecture ($\boldsymbol{\theta}$) and the input data ($\boldsymbol{x}_i$). The underlying rationale is that overparameterized DNNs are sufficiently expressive to learn data embeddings that are primarily driven by minimizing the training loss, rather than being strictly constrained by the specific architecture. The UFM has been widely adopted and validated, e.g., (Yang et al., 2017; Mixon et al., 2020; Fang et al., 2021; Zhu et al., 2021; Zhao et al., 2024).
Formally, we abstract the DNN as a **bilinear model** where the logits $\boldsymbol{L} \in \mathbb{R}^{k\times n}$ are defined as $\boldsymbol{L} := \boldsymbol{W}\boldsymbol{H}$. Here, $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ represents the network's classifiers (analogous to the last-layer weights), and $\boldsymbol{H} \in \mathbb{R}^{d\times n}$ is a matrix comprising the trainable embeddings $\boldsymbol{h}_i, i \in [n]$ for each example. Both $\boldsymbol{W}$ and $\boldsymbol{H}$ are parameters to be optimized during training.

Loss simplification. Instead of the cross-entropy loss, we employ the squared-loss. This simplification is a common practice in theoretical analyses, as squared-loss generally leads to more interpretable and analytically tractable learning dynamics. Specifically, we use squared-loss with a **simplex-encoded label (SEL) matrix** $\boldsymbol{Z} := (\mathbb{I}_k - \frac{1}{k}\mathbb{1}_k\mathbb{1}_k^\top)\boldsymbol{Y}$. This matrix is a centered version of the one-hot encoding matrix $\boldsymbol{Y} \in \{0,1\}^{k\times n}$ of the data. The centering accounts for the property of cross-entropy to yield centered logits, i.e., such that $\mathbf{1}_k^\top \boldsymbol{L} = \mathbf{0}$ (Thrampoulidis et al., 2022). Viewed this way, the simplification to squared-loss can alternatively be considered a first-order approximation of the cross-entropy loss.

Overparameterization. To model overparameterization, within our bilinear model, we set hidden dimension $d \geq k$. This ensures model parameters exist that can perfectly interpolate the labels, meaning $\boldsymbol{L} = \boldsymbol{Z}$.

STEP-Imbalance. For concreteness, we assume k classes, where the first $k/2$ are **majorities** and the remaining $k/2$ are **minorities**. Within each group, all classes have an equal number of examples. The number of samples in majorities is R times the number of samples in minorities, where R is the **imbalance ratio**. For simplicity and without compromising the generality of our findings, we set $k = 4$ for our detailed discussion. See Fig. 2(a) the 1-hot label matrix $\boldsymbol{Y}$.

Remark 3.1. We view this model as the minimal canonical model capable of transparently justifying how reweighting balances feature learning rates during training. It is minimal because it essentially describes the dynamics of a two-layer linear neural network with standard basis inputs. The model is canonical in that it avoids imposing explicit assumptions on the geometry of the abstracted input data—unlike a simpler linear model would necessitate. Instead, its abstraction is based on assuming an overparameterized and sufficiently expressive architecture.

3.2. A key Spectral Interpretation

The SSM isolates the encoding of the imbalanced data structure within the spectral components of the SEL matrix, $\boldsymbol{Z}$, in a way that directly reveals how the model prioritizes learning about majorities over minorities. Let $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be the SVD of the SEL matrix. It can be formally shown that $\boldsymbol{\Sigma}$ has $k-1$ non-zero singular values, which are categorized into *three distinct levels* such that the largest and smallest values each have a multiplicity of $k/2-1$, while the middle value has a multiplicity of one. These three distinct levels correspond to three groups of principal components each associated with features that the model must separately learn to distinguish: (i) majority classes from each other (**`maj-maj`** feature), (ii) majority classes from minority classes (**`maj-min`** feature), and (iii) minority classes from each other (**`min-min`** feature). These features are directly encoded as singular vectors of $\boldsymbol{Z}$.

For illustration, consider the case $k = 4$. Observe in Fig. 2(b) the left singular matrix $\boldsymbol{U}$, where each column represents one

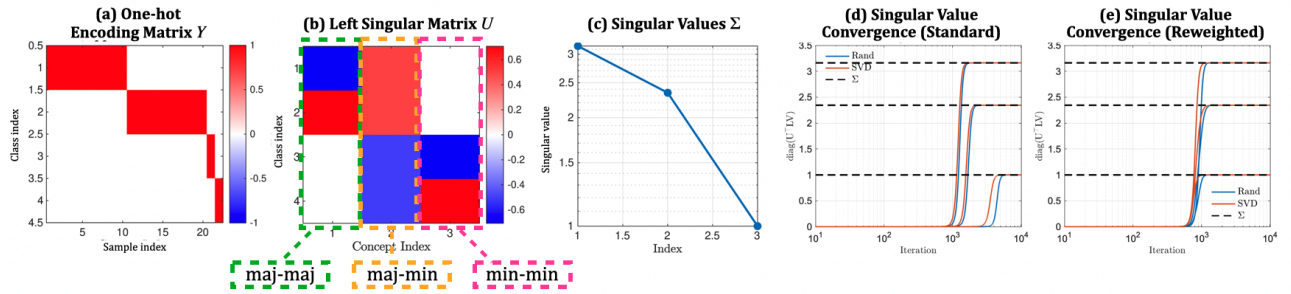

Figure 2. **Spectral analysis of the SSM with 4-class STEP-imbalance (imbalance ratio $R = 10$).** (a) One-hot label matrix $\mathbf{Y}$ for two majority(top) and two minority(bottom) classes. (b) Left singular vectors $\boldsymbol{U}$ of the SEL matrix $\boldsymbol{Z} = (I - \frac{1}{k}\mathbb{1}\mathbb{1}^\top)\mathbf{Y}$ separate majority-majority (green), majority-minority (orange), and minority-minority (pink) distinctions. (c) Singular values show the ordering of semantic importance: majority features dominate. (d) Under standard training, features are learned sequentially. (e) Reweighting equalizes learning speeds, flattening the spectrum and enabling simultaneous semantics acquisition. (Blue is random and orange is spectral initialization.)

of these features. We can interpret the sign of each of the k entries in a singular vector $\boldsymbol{u}_j$ as an indicator of how strongly the respective class is associated with that principal component: if $\boldsymbol{u}_j[c] > 0$ or $\boldsymbol{u}_j[c] < 0$, it signifies that class c aligns with or opposes this component, respectively; if $\boldsymbol{u}_j[c] = 0$, the class is neutral to that component. We thus find **three principal components, corresponding to distinct features, ordered by the magnitude of their respective singular values:**

1. **`maj-maj`**: The first principal component ($j = 1$) is neutral to minorities and distinguishes between the two majorities (indicated by the opposing signs of $\boldsymbol{u}_1[1]$ and $\boldsymbol{u}_1[2]$).
2. **`maj-min`**: The second principal component ($j = 2$) distinguishes between majorities and the minorities (indicated by the opposing signs of $\boldsymbol{u}_2[1], \boldsymbol{u}_2[2]$ and $\boldsymbol{u}_2[3], \boldsymbol{u}_2[4]$).
3. **`min-min`**: The last principal component ($j = 3$) is neutral to majorities and distinguishes between the two minorities (indicated by the opposing signs of $\boldsymbol{u}_3[3]$ and $\boldsymbol{u}_3[4]$).

3.3. Reweighting Changes the Learning Dynamics Eliminating Preferential Learning Against Minorities

We now leverage our SSM to illuminate how reweighting fundamentally alters the learning dynamics. Specifically, we compare the training dynamics under gradient descent for vanilla ERM versus loss reweighting:

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{H}) := \frac{1}{n}\sum\nolimits_{i\in[n]} \|\boldsymbol{z}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2 \qquad \texttt{versus} \qquad \mathcal{L}_{\mathrm{RW}}(\boldsymbol{W}, \boldsymbol{H}) := \frac{1}{n}\sum\nolimits_{i\in[n]} \omega_{y_i} \cdot \|\boldsymbol{z}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2$$

where the weights ω_{y_i} are set inversely proportional to the frequency of the respective class $y_i \in [k]$. Concretely, if the class y_i of the i-th example is a majority, then $\omega_{y_i} \propto \sqrt{(R+1)V/(2R)}$, otherwise, for a minority class, $\omega_{y_i} \propto \sqrt{(R+1)V/2}$.[2] For training, we use gradient descent with small random initialization and small step-size.

Fig. 2(d) and 2(e) tracks the rates at which the singular values of the logits $\boldsymbol{L}_t$ reach their terminal values in both scenarios (vanilla ERM vs. reweighting) across training iterations t. We observe the following about vanilla ERM versus reweighting:

1. **Terminal Phase Equivalence:** Due to overparameterization ($d \geq k$), there is *no* difference in the terminal phase. In both cases, the singular values of $\boldsymbol{L}_t$ converge to the singular values of the SEL matrix $\boldsymbol{Z}$.
2. **Preferential learning in vanilla ERM hurts minorities:** With vanilla ERM ($\mathcal{L}$), the model learns the three singular values in a highly preferential order, with those corresponding to the larger singular values of $\boldsymbol{Z}$ being learned first.
3. **Reweighting restores balanced Learning:** In contrast, loss reweighting ($\mathcal{L}_{\mathrm{RW}}$) eliminates this preferential ordering, leading to all singular values being learned at approximately equal rates.

Recalling that singular values directly correspond to distinct features, our SSM thus captures the **early training benefit of reweighting:** While vanilla ERM $\mathcal{L}$ learns **`maj-maj`**, **`maj-min`**, **`min-min`** features in that specific order (thereby favoring majority learning and delaying minority learning until the end of training), reweighted ERM $\mathcal{L}_{\mathrm{RW}}$ learns all features

[2]Note that this is equivalent to using weights $\omega_{y_i} \propto 1/\hat{\pi}_{y_i}^{\gamma}$ with $\gamma = 0.5$. This particular choice of γ is motivated by our theoretical analysis in Section 4 and, interestingly, aligns with empirical findings from prior work (Cao et al., 2019; Menon et al., 2020; Behnia et al., 2023; Kini et al., 2021; Ye et al., 2020) who report $\gamma = 1/2$ as a robust choice in practice.

at the same rate, thus enabling accurate classification of minorities much earlier. Note the direct one-to-one correspondence of these behaviors to what we observed empirically in the DNN experiment presented in Section 2.

4. Analysis

By identifying a connection of the square-loss UFM with the framework of Saxe et al. (2013), we have closed-form dynamics for gradient flow on $\mathcal{L}$ with appropriate spectral initialization (see Prop. C.1). Leveraging closed-form expressions for the spectral factors of the SEL matrix, we further establish closed-form formulas for $\mathcal{L}_{\mathrm{RW}}$ in Thm. C.2. These closed-form formulas precisely verify the three findings from the analysis presented in the previous section. In summary, although the training trajectories for both vanilla ERM and reweighting converge to the same final values at the end of training, a critical difference lies in the *effective learning window* $\Delta T := (T_{\max} - T_{\min})/T_{\min}$ from time $T_{\min}$ when the model learns the first principal component until $T_{\max}$ when it has learned all principal components. For vanishing initialization, we show that $\Delta T_{\mathrm{ERM}} = \sqrt{R} - 1$ grows uboundedly with the imbalance ratio R, whereas $\Delta T_{\mathrm{RW}} = \sqrt{2}\frac{\sqrt{R+1}}{\sqrt{R}+1}$ is upper bounded by $\sqrt{2}$ irrespective of imbalance R. See App. C for details.

5. Future Work

Our model already rather effectively captures the early-training impacts of reweighting under overparameteration. In future work, we aim to address limitations by extending the analysis beyond squared loss and also to test-time dynamics.

6. Acknowledgments

This work was funded by the NSERC Discovery Grant No. 2021-03677, the Alliance Grant ALLRP 581098-22, and an Alliance Mission Grant. The authors also acknowledge use of the Sockeye cluster by UBC Advanced Research Computing.

References

Tina Behnia, Ke Wang, and Christos Thrampoulidis. On how to avoid exacerbating spurious correlations when models are overparameterized. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 121–126. IEEE, 2022.

Tina Behnia, Ganesh Ramachandra Kini, Vala Vakilian, and Christos Thrampoulidis. On the implicit geometry of cross-entropy parameterizations for label-imbalanced data. In *International Conference on Artificial Intelligence and Statistics*, pages 10815–10838. PMLR, 2023.

Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881. PMLR, 2019.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pages 1567–1578, 2019.

Ahmet Demirkaya, Jiasi Chen, and Samet Oymak. Exploring the role of loss functions in multiclass classification. In *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5, 2020. doi: 10.1109/CISS48834.2020.1570627167.

Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences*, 118(43), 2021.

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.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR, 2018a.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in Neural Information Processing Systems*, 31:9461–9471, 2018b.

XY Han, Vardan Papyan, and David L Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

Wanli Hong and Shuyang Ling. Neural collapse for unconstrained feature model under cross-entropy loss with imbalanced data. *arXiv preprint arXiv:2309.09725*, 2023.

Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322*, 2020.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pages 1772–1798. PMLR, 2019.

Byungju Kim and Junmo Kim. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8:81674–81685, 2020. doi: 10.1109/ACCESS.2020.2991231.

Ganesh Ramachandra Kini, Orestis Paraskevas, Samet Oymak, and Christos Thrampoulidis. Label-imbalanced and group-sensitive classification under overparameterization. *Advances in Neural Information Processing Systems*, 34:18970–18983, 2021.

Kuo-Wei Lai and Vidya Muthukumar. Sharp analysis of out-of-distribution error for "importance-weighted" estimators in the overparameterized regime. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 3701–3706. IEEE, 2024.

Mingchen Li, Xuechen Zhang, Christos Thrampoulidis, Jiasi Chen, and Samet Oymak. Autobalance: Optimized loss functions for imbalanced data. *Advances in Neural Information Processing Systems*, 34:3163–3177, 2021.

Haixia Liu. The exploration of neural collapse under imbalanced data. *arXiv preprint arXiv:2411.17278*, 2024.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *arXiv preprint arXiv:2011.11619*, 2020.

Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):11, 2022.

Eliav Mor and Yair Carmon. An analytical model for overparameterized learning under class imbalance. *arXiv preprint arXiv:2503.05289*, 2025.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

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.

Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Peter Súkeník, Marco Mondelli, and Christoph H Lampert. Deep neural collapse is provably optimal for the deep unconstrained features model. *Advances in Neural Information Processing Systems*, 36:52991–53024, 2023.

Christos Thrampoulidis, Ganesh R Kini, Vala Vakilian, and Tina Behnia. Imbalance trouble: Revisiting neural-collapse geometry. *arXiv preprint arXiv:2208.05512*, 2022.

Tom Tirer and Joan Bruna. Extended unconstrained features model for exploring deep neural collapse. *arXiv preprint arXiv:2202.08087*, 2022.

Ke Alexander Wang, Niladri S Chatterji, Saminul Haque, and Tatsunori Hashimoto. Is importance weighting incompatible with interpolating classifiers? *arXiv preprint arXiv:2112.12986*, 2021.

Zitai Wang, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. A unified generalization analysis of re-weighting and logit-adjustment for imbalanced learning. *Advances in Neural Information Processing Systems*, 36: 48417–48430, 2023.

Monica Welfert, Nathan Stromberg, and Lalitha Sankar. Theoretical guarantees of data augmented last layer retraining methods. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 581–586. IEEE, 2024.

Da Xu, Yuting Ye, and Chuanwei Ruan. Understanding the role of importance weighting for deep learning. *arXiv preprint arXiv:2103.15209*, 2021.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. Breaking the softmax bottleneck: A high-rank rnn language model. *arXiv preprint arXiv:1711.03953*, 2017.

Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning, 2020.

Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. Understanding why generalized reweighting does not improve over erm. *arXiv preprint arXiv:2201.12293*, 2022.

Yize Zhao, Tina Behnia, Vala Vakilian, and Christos Thrampoulidis. Implicit geometry of next-token prediction: From language sparsity patterns to model representations. *arXiv preprint arXiv:2408.15417*, 2024.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34, 2021.

A. Background on Loss Reweighting

A.1. Statistical Optimality in the Classical Regime

Loss reweighting is theoretically grounded, being statistically optimal in the population limit for optimizing balanced accuracy, which weighs class-conditional accuracies equally rather than by class frequencies. Concretely, let π_c be the prior probability of class $c \in [k]$ in a k-class classification dataset. The standard error of a hypothesis h is $\mathcal{R}(h) := \sum_{c\in[k]} \pi_c \mathcal{R}_c(h) = \mathbb{E}_{c\sim P(c), \boldsymbol{x}\sim P(\boldsymbol{x}|c)}[\mathbf{1}[h(\boldsymbol{x}) \neq c]]$, where $\mathcal{R}_c(h) = \mathbb{E}_{\boldsymbol{x}\sim P(\boldsymbol{x}|c)}[\mathbf{1}[h(\boldsymbol{x}) \neq c]]$ is the class-conditional error. In practice, given n samples $(\boldsymbol{x}_i)_{i=1}^n$ with labels $y_i \in [k]$, we minimize the empirical risk

$$\mathcal{L}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(h(\boldsymbol{x}_i), y_i\right), \tag{2}$$

where ℓ denotes a differentiable proxy to the zero-one loss. The balanced error is $\mathcal{R}_{\text{bal}}(h) := \frac{1}{k}\sum_{c\in[k]} \mathcal{R}_c(h)$, with empirical counterpart $\mathcal{L}_{\text{bal}}(h) = \frac{1}{k}\sum_{c\in[k]} \frac{1}{n_c}\sum_{i:y_i=c} \ell\left(h(\boldsymbol{x}_i), c\right)$, where $n_c = |\{i : y_i = c\}|$. This can be rewritten as a reweighted loss:

$$\mathcal{L}_{\text{reweight}}(h) = \frac{1}{n}\sum_{i=1}^{n} \omega_i \cdot \ell(h(\boldsymbol{x}_i), y_i), \tag{3}$$

where $\omega_i = 1/(n_{y_i}/n)$ is the inverse empirical frequency of each class. Under mild assumptions, this re-weighted loss is statistically optimal when $n \to \infty$ and the hypothesis space is bounded.

A.2. Nuanced Picture in the Overparameterization Regime

However, modern machine learning challenges the (classical) population analysis: neural networks operate in high-dimensional regimes where the hypothesis-space dimension exceeds the training sample count. In such overparameterized settings, classical methods often lose their optimality (). Indeed, several works (Byrd and Lipton, 2019; Kim and Kim, 2020) have identified weighted cross-entropy's failure to substantially improve balanced accuracy of well-trained overparameterized deep neural networks on imbalanced datasets.

Investigating this behavior theoretically, Sagawa et al. (2020) and Kini et al. (2021) attribute this failure to the implicit optimization bias of gradient-based optimizers used to minimize the loss during training. A long line of research (Gunasekar et al., 2018b;a; Soudry et al., 2018; Ji and Telgarsky, 2019; Lyu and Li, 2019) has supported the idea that empirical risk minimization of overparameterized networks—where many minimizers of the loss can exist—is empirically successful precisely because first-order gradient optimizers are biased toward solutions that maximize the margin over the training data among all train loss minimizers. Applied to weighted cross-entropy, Sagawa et al. (2020) and Kini et al. (2021) leverage this type of analysis to demonstrate that loss reweighting has no effect on the implicit optimization bias: irrespective of the choice of weights, when trained long enough, the configuration chosen by gradient-based optimization applied to weighted cross-entropy is the same as vanilla cross-entropy.

Kini et al. (2021) identified that in this overparameterized regime reweighting still works, but should be done differently than suggested by Eq. (3) and should instead be applied to the logits. See also related (Cao et al., 2019; Menon et al., 2020; Xu et al., 2021; Welfert et al., 2024) and follow-up works e.g., (Behnia et al., 2022; 2023; Wang et al., 2021; Lai and Muthukumar, 2024; Mor and Carmon, 2025; Wang et al., 2023; Zhai et al., 2022).

Despite being very insightful and leading to practical modifications, there are two limitations in the above implicit bias analysis: First, the implicit optimization bias analysis provides explicit characterizations of the max-margin classifiers to which optimizers converge only in limited settings (even for two-layer linear networks, it has not been rigorously shown whether gradient descent converges to the global minimizer of the non-convex max-margin optimization). Second, and perhaps more importantly, the implicit bias toward a max-margin classifier manifests very late in training: even in linear models, convergence to the max-margin solution with gradient descent is exponentially slow (convergence can be accelerated with normalization, but still requires many iterations).

On the other hand, it has been empirically observed that loss reweighting can actually be beneficial when either used with early stopping or as a complementary technique to modern loss variants optimized for the overparameterized regime, leading to empirical speed-ups and optimization boosts in early training iterations (Cao et al., 2019; Xu et al., 2021; Li et al., 2021).

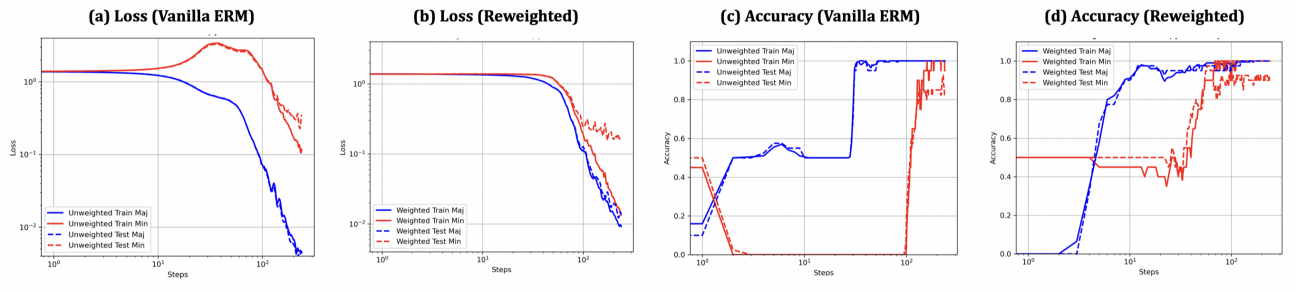

Figure 3. Training dynamics of majority vs. minority classes on an imbalanced 4-class MNIST task ($R = 10$, 2 majority, 2 minority). **(a)** Under *vanilla ERM*, minority loss increases at early stage and remains higher in training, while majority loss quickly drops. **(b)** Under *reweighted ERM*, both majority and minority losses decrease together, indicating balanced optimization. **(c)** Vanilla ERM results in delayed learning for minorities, as majority accuracy increases early and minority accuracy only improves after 100 steps. **(d)** Reweighting leads to a relatively synchronized accuracy gain for both groups, with minority test accuracy improving much earlier.

B. Additional Experimental Details on Motivating Example

B.1. Loss Trajectories

Figure 3 (a)-(b) shows the loss curves for minority and majority classes.

B.2. Accuracy Dynamics

Figure 3 (c)-(d) plots accuracies.

B.3. Confusion Matrix Progression on Test Data

To further illustrate the difference in generation ability between standard and reweighted training, we visualize the evolution of test-set confusion matrices over training steps (Figure 4). Each matrix shows class-level prediction accuracy at a given step.

These trends validate our spectral hypothesis: *vanilla CE follows the ordering of singular values of the label matrix—favoring majority-majority features—while Re-weighted flattens this spectrum and equalizes learning rates across all concept directions.*

B.4. Sensitivity to Digit Selection

We observe that the specific digits chosen as majority vs. minority classes can influence learning dynamics. For instance, using `maj` = (2,3), `min` = (0,1) leads to unusually fast minority learning under reweighted loss—minorities are correctly classified from the start. In contrast, reversing the roles (`maj` = (0,1), `min` = (2,3)) delays minority learning.

This asymmetry likely reflects differences in digit difficulty (e.g., 0 and 1 are easier to distinguish than 2 and 3). Importantly, across all configurations tested, loss reweighting consistently accelerates minority learning relative to standard loss, even if the absolute ordering varies.

C. Analysis Details

C.1. Background on Vanilla ERM

Consider the square-loss UFM:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \left\{ \frac{1}{2} \sum_{i\in[n]} \|\boldsymbol{z}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2 = \frac{1}{2}\|\boldsymbol{Z} - \boldsymbol{W}\boldsymbol{H}\|^2 \right\}. \tag{4}$$

which fits logits $\boldsymbol{W}\boldsymbol{H}$ to the centered sparsity matrix $\boldsymbol{Z}$. Although CE loss is more common in practice, we note that square-loss has shown competitive performance to CE minimization in various settings (Hui and Belkin, 2020; Demirkaya

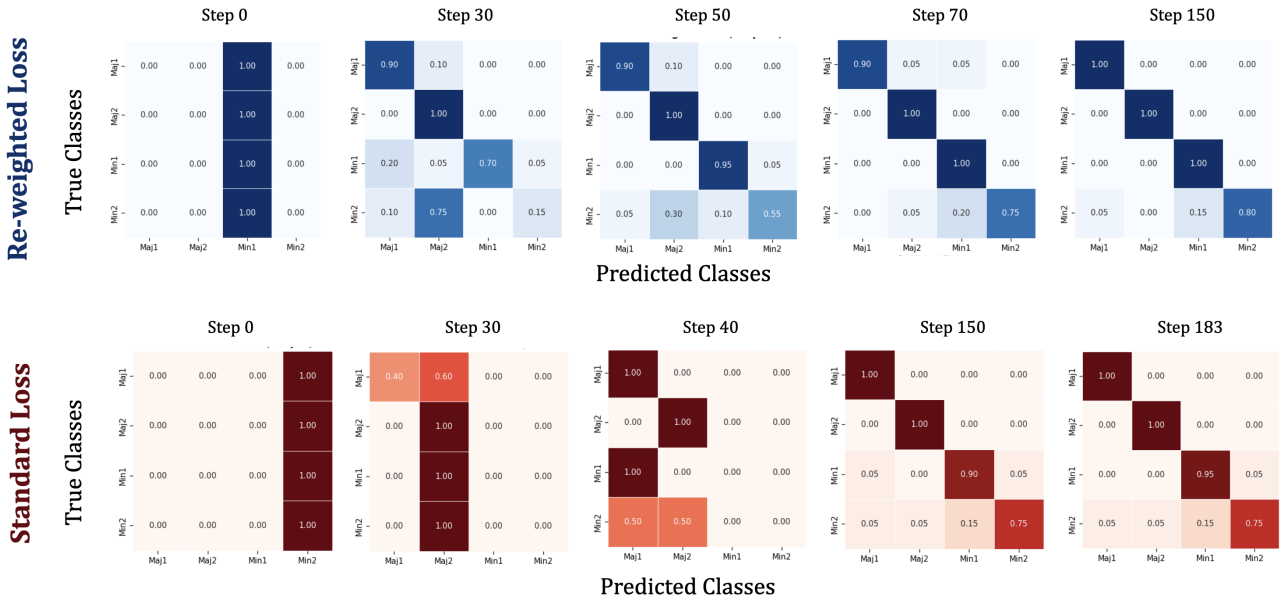

Figure 4. Confusion matrix progression on test data under reweighted (top row) vs. standard (bottom row) cross-entropy loss. Each matrix shows predictions across 4 classes (2 majority, 2 minority) at selected training steps. Under standard loss, the model first learns majority classes, with minority learning delayed. Reweighting enables earlier and more balanced classification across all classes.

et al., 2020). The neural-collapse literature has extensively studied Eq. (4) primarily in the balanced case (e.g., (Mixon et al., 2020; Han et al., 2021; Súkeník et al., 2023; Tirer and Bruna, 2022)) but recently also for imbalanced data (e.g., (Liu, 2024; Hong and Ling, 2023)). Most works focus on global minima of regularized UFM, with less attention to unregularized cases or training dynamics. While some landscape analyses provide partial answers about global convergence Mixon et al. (2022); Han et al. (2021), they are limited to regularized cases and do *not* characterize dynamics. For example, (Han et al., 2021)'s analysis of the 'central path' in balanced one-hot cases—these results relies on approximations. Thus, a significant gap remains in understanding UFM training dynamics, even for simple balanced one-hot data with square loss.

By interpreting the UFM with square loss in Eq. (4) as a two-layer linear network with orthogonal inputs, we identify a connection to Saxe et al. (2013); Gidel et al. (2019)'s analysis, that to the best of our knowledge has thus far remained unexplored in the neural-collapse literature. Saxe et al. (2013) provide explicit characterization of gradient descent dynamics (with small initialization) for square-loss UFM. The key insight in adopting their results, is rewriting (4) as $\sum_{i\in[n]} \|\boldsymbol{z}_i - \boldsymbol{W}\boldsymbol{H}\boldsymbol{e}_i\|^2$ with *orthogonal* inputs $\boldsymbol{e}_i \in \mathbb{R}^m$. This enables direct application of their result, originally stated in (Saxe et al., 2013) and formalized in (Gidel et al., 2019). For completeness, we state this here in our setting and terminology as a proposition below.

Proposition C.1 ((Saxe et al., 2013; Gidel et al., 2019))**.** *Consider gradient flow (GF) dynamics for minimizing the square-loss UFM* (4)*. Recall the SVD* $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$*. Assume weight initialization*

$$\boldsymbol{W}(0) = e^{-\delta}\boldsymbol{U}\mathbf{R}^\top \quad \textit{and} \quad \boldsymbol{H}(0) = e^{-\delta}\mathbf{R}\boldsymbol{V}^\top$$

for some partial orthogonal matrix $\mathbf{R} \in \mathbb{R}^{d\times r}$ *(*$\mathbf{R}^\top\mathbf{R} = \mathbb{I}_r$*) and initialization scale* $e^{-\delta}$*. Then the iterates* $\boldsymbol{W}(t), \boldsymbol{H}(t)$ *of GF are as follows:*

$$\boldsymbol{W}(t) = \boldsymbol{U}\sqrt{\boldsymbol{\Sigma}}\sqrt{\boldsymbol{A}(t)}\mathbf{R}^\top \qquad \textit{and} \qquad \boldsymbol{H}(t) = \mathbf{R}\sqrt{\boldsymbol{\Sigma}}\sqrt{\boldsymbol{A}(t)}\boldsymbol{V}^\top \tag{5}$$

for $\boldsymbol{A}(t) = \mathrm{diag}(a_1(t), \ldots, a_r(t))$ *with*

$$a_i(t) = \frac{1}{1 + (\sigma_i e^{2\delta} - 1)e^{-2\sigma_i t}},\ i \in [r]. \tag{6}$$

Moreover, the time-rescaled factors $a_i(\delta t)$ converge to a step function as $\delta \to \infty$ (limit of vanishing initialization):

$$a_i(\delta t) \to \frac{1}{1+\sigma_i}\mathbb{1}[t = T_i] + \mathbb{1}[t > T_i], \tag{7}$$

where $T_i = 1/\sigma_i$ and $\mathbb{1}[A]$ is the indicator function for event A. Thus, the i-th component is learned at time T_i inversely proportional to σ_i.

Proof. After having set up the analogy of our setting to that of (Saxe et al., 2013; Gidel et al., 2019), this is a direct application of (Gidel et al., 2019, Thm. 1). Specifically, this is made possible in our setting by: (i) interpreting the UFM with square-loss in (4) as a two-layer linear network (ii) recognizing that the covariance of the inputs (which here are standard basis vectors $\boldsymbol{e}_j, j \in \mathbb{R}^m$) is the identity matrix, hence the (almost) orthogonality assumption (see (Saxe et al., 2013) and (Gidel et al., 2019, Sec. 4.1)) holds. □

This result requires initializing $\boldsymbol{W}, \boldsymbol{H}$ in a way that makes them aligned with the SVD factors of the SEL matrix. While this might appear as a strong assumption, (Saxe et al., 2013; 2019) conjectured and verified experimentally that the characterization remains qualitatively accurate under small random initialization. Our experiments with the SEL matrix confirm this - Fig. 5 (middle row) shows the singular values of the logit matrix during training closely follow the predicted exponential trend in Eq. (16). This reveals that dominant singular factors, corresponding to primary semantic concepts, are learned first. In the limit $t \to \infty$, the theorem shows convergence to:

$$\boldsymbol{W}(t) \to \boldsymbol{W}_\infty := \boldsymbol{U}\sqrt{\boldsymbol{\Sigma}}\mathbf{R}^\top \qquad \text{and} \qquad \boldsymbol{H}(t) \to \boldsymbol{H}_\infty := \mathbf{R}\sqrt{\boldsymbol{\Sigma}}\boldsymbol{V}^\top, . \tag{8}$$

This aligns with (Thrampoulidis et al., 2022)'s regularization-path analysis of UFM with CE loss, where normalized quantities converge as regularization goes to zero.

C.2. Controlling the rate of learning via Reweighting

Consider minimizing the following weighted version of (4):

$$\min_{\boldsymbol{W},\boldsymbol{H}} \left\{ \frac{1}{2}\sum_{i\in[n]} \omega_i \|\boldsymbol{z}_i - \boldsymbol{W}\boldsymbol{h}_i\|_2^2 = \frac{1}{2}\left\|(\boldsymbol{Z} - \boldsymbol{W}\boldsymbol{H})\,\boldsymbol{\Omega}^{1/2}\right\|_F^2 \right\}, \tag{9}$$

where $\boldsymbol{\Omega} = \mathrm{diag}([\omega_1, \ldots, \omega_m])$ is a diagonal matrix of weights, one for each context. We consider the STEP-imbalanced one-hot classification setting described. Concretely, let the one-hot label matrix be

$$\boldsymbol{Y} = \begin{bmatrix} \mathbb{I}_{V/2} \otimes \mathbb{1}_R^\top & \mathbf{0}_{V/2\times V/2} \\ \mathbf{0}_{V/2\times RV/2} & \mathbb{I}_{V/2} \end{bmatrix} \tag{10}$$

where R is the imbalance ratio and without loss of generality we assumed that the first $V/2$ classes are majorities and that minorities have 1 example each. Here, the choice of 1 sample per minority is done without loss of generality and just to maintain simplicity in the formulas. Thus, the total number of examples is $n = RV/2 + V/2 = (R+1)V/2$. Recall that $\boldsymbol{Z} = (\mathbb{I}_V - \frac{1}{V}\mathbb{1}_V\mathbb{1}_V^\top)\boldsymbol{Y}$.

Recall that $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Thrampoulidis et al. (2022, Lem. A.3) derives the spectral components in closed form as follows:

$$\boldsymbol{\Sigma} = \mathrm{diag}(\begin{bmatrix}\sqrt{R}\mathbb{1}_{V/2-1} & \sqrt{(R+1)/2} & \mathbb{1}_{V/2-1}\end{bmatrix}) \tag{11}$$

$$\boldsymbol{U} = \begin{bmatrix} \mathbb{F} & -\sqrt{\frac{1}{V}}\mathbb{1} & \mathbf{0} \\ \mathbf{0} & \sqrt{\frac{1}{V}}\mathbb{1} & \mathbb{F} \end{bmatrix} \in \mathbb{R}^{V\times(V-1)},$$

and

$$\boldsymbol{V}^\top = \begin{bmatrix} \sqrt{\frac{1}{R}}\mathbb{F}^\top \otimes \mathbb{1}_R^\top & \mathbf{0} \\ -\sqrt{\frac{2}{(R+1)V}}\mathbb{1}_{RV/2}^\top & \sqrt{\frac{2}{(R+1)V}}\mathbb{1}_{V/2}^\top \\ \mathbf{0} & \mathbb{F}^\top \end{bmatrix} \in \mathbb{R}^{(V-1)\times m}.$$

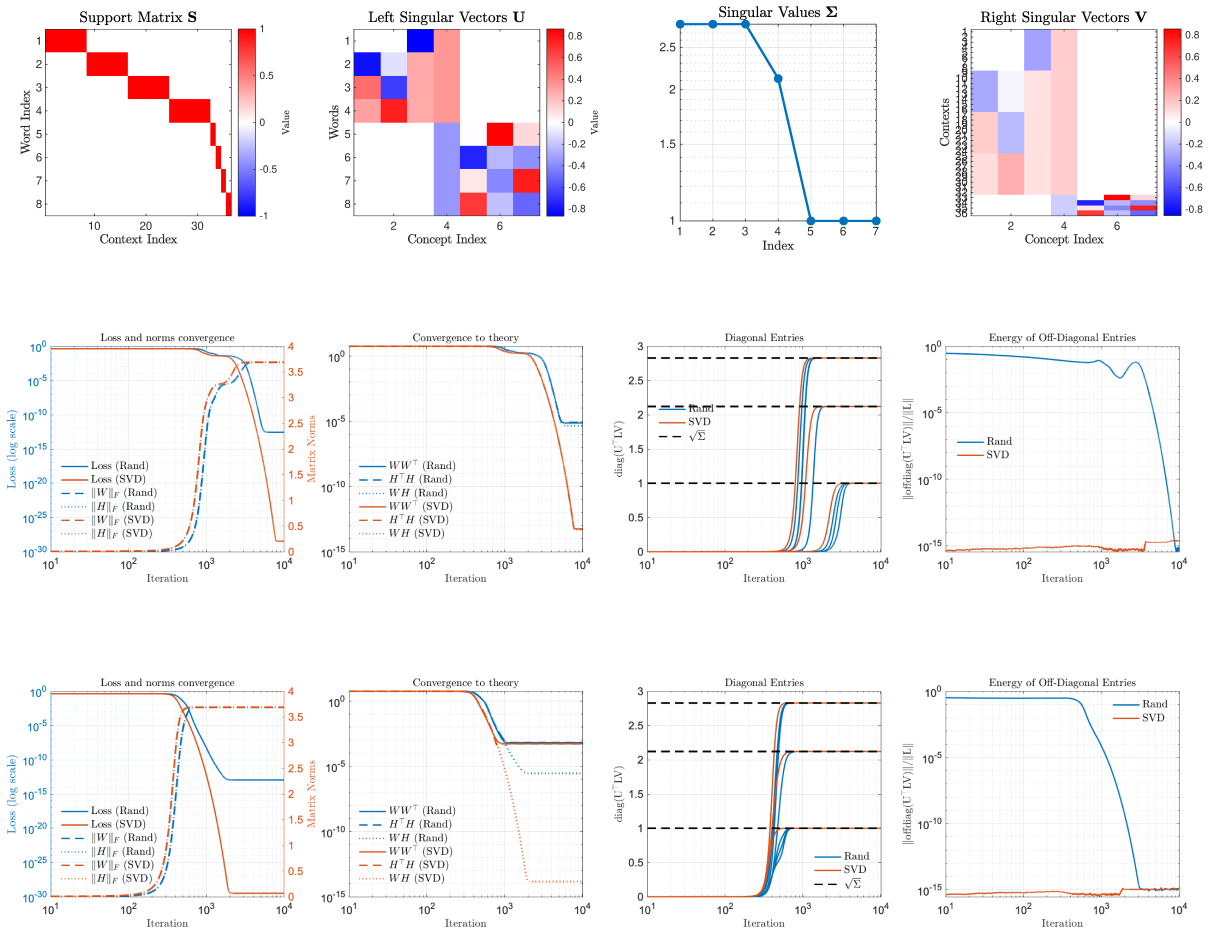

Figure 5. **(Top)** One-hot encoding matrix and SVD factors of SEL matrix for STEP-imbalanced data. **(Middle)** Training dynamics of GD minimization of UFM with square loss (Eq. (4). for two initializations: (i) SVD: initialize $\boldsymbol{W}$ and $\boldsymbol{H}$ as per Thm. C.1 for $\delta = 8$. (ii) Rand: intialize $\boldsymbol{W}$ and $\boldsymbol{H}$ random Gaussian scaled to match the norm of SVD initialization. Dynamics with the two initialization are shown in orange (SVD) and blue (Rand), respectively. Qualitatively the behavior is similar. With vanilla ERM, SVD factors are learned in the order of their singular values, see Fig. 6 for a more general synthetic setting. **(Bottom)** Training dynamics of GD minimization of UFM with weighted square loss (Eq. (9)) with weights as in Eq. (12). Note that thanks to the weighting all singular factors are now learned at approximately the same rates.

Above, $\mathbb{F} \in \mathbb{R}^{V/2\times(V/2-1)}$ is an orthonormal basis of the subspace orthogonal to $\mathbb{1}_{V/2}$.

We now set the weights inversely proportional to the square-root of the respective class frequency, i.e. the weight for the first $RV/2$ majority samples is $\sqrt{n/R} = \sqrt{(R+1)V/(2R)}$, while for the minorities is $\sqrt{R}$ times larger, i.e., $\sqrt{n} = \sqrt{(R+1)V/2}$. The relative rates of learning of majority/minority features does not change if we uniformly scale all weights by a constant. Thus, for simplicity and without affecting our analysis, we drop the factor $\sqrt{V/2}$, and, in matrix form, choose

$$\boldsymbol{\Omega} := \sqrt{R+1} \cdot \mathrm{diag}\Big(\Big[\sqrt{\tfrac{1}{R}}\,\mathbb{1}_{RV/2}^\top \quad \mathbb{1}_{V/2}^\top\Big]\Big)\,. \tag{12}$$

Direct calculation yields the following:

$$\boldsymbol{V}^\top \boldsymbol{\Omega} \boldsymbol{V} = \sqrt{R+1} \begin{bmatrix} \frac{1}{R}\mathbb{F}^\top \otimes \mathbb{1}_R^\top & \mathbf{0} \\ -\sqrt{\frac{2}{R(R+1)V}}\mathbb{1}_{RV/2}^\top & \sqrt{\frac{2}{(R+1)V}}\mathbb{1}_{V/2}^\top \\ \mathbf{0} & \mathbb{F}^\top \end{bmatrix} \begin{bmatrix} \sqrt{\frac{1}{R}}\mathbb{F} \otimes \mathbb{1}_R & -\sqrt{\frac{2}{(R+1)V}}\mathbb{1}_{RV/2} & \mathbf{0} \\ \mathbf{0} & \sqrt{\frac{2}{(R+1)V}}\mathbb{1}_{V/2} & \mathbb{F} \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{\frac{R+1}{R}}\,\mathbb{I}_{V/2-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\sqrt{R}+1}{\sqrt{R+1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{R+1}\,\mathbb{I}_{V/2-1} \end{bmatrix} =: \boldsymbol{\Lambda}. \tag{13}$$

where in the last equation, we defined the diagonal matrix $\boldsymbol{\Lambda}$ with entries:

$$\lambda_i := \begin{cases} \sqrt{\frac{R+1}{R}} & i \in [V/2-1] \\ \frac{\sqrt{R}+1}{\sqrt{R+1}} & i = V/2 \\ \sqrt{R+1} & i = V/2+1, \dots, V-1 \end{cases} \tag{14}$$

This calculation is handy in the proof of the following theorem which is our main theoretical result. The proof shows that $\boldsymbol{\Lambda}$ is the effective weight matrix applied to the spectrum of the label matrix.

Theorem C.2. *Consider gradient flow (GF) dynamics for minimizing the **weighted** square-loss UFM (9) with weight matrix as shown in Eq. (12) in an R-STEP-imbalanced setting. Assume same parameter initialization as in Proposition C.1, i.e., $\boldsymbol{W}(0) = e^{-\delta}\boldsymbol{U}\mathbf{R}^\top$, $\boldsymbol{H}(0) = e^{-\delta}\mathbf{R}\boldsymbol{V}^\top$, for partial orthogonal matrix $\mathbf{R} \in \mathbb{R}^{d\times(V-1)}$, initialization scale $e^{-\delta}$ and SVD $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Finally, consider the **effective weights** $\lambda_i, i \in [V-1]$ defined in Eq. (14). Then the iterates $\boldsymbol{W}(t), \boldsymbol{H}(t)$ of GF evolve as follows:*

$$\boldsymbol{W}(t) = \boldsymbol{U}\sqrt{\boldsymbol{\Sigma}}\sqrt{\boldsymbol{B}(t)}\mathbf{R}^\top \qquad \text{and} \qquad \boldsymbol{H}(t) = \mathbf{R}\sqrt{\boldsymbol{\Sigma}}\sqrt{\boldsymbol{B}(t)}\boldsymbol{V}^\top \tag{15}$$

for $\boldsymbol{B}(t) = \mathrm{diag}(\beta_1(t), \dots, \beta_{V-1}(t))$ with

$$\beta_i(t) = \frac{1}{1 + (\sigma_i e^{2\delta} - 1)e^{-2\sigma_i\lambda_i t}},\ i \in [V-1]. \tag{16}$$

Moreover, the time-rescaled factors $\beta_i(\delta t)$ converge to a step function as $\delta \to \infty$:

$$\beta_i(\delta t) \to \frac{1}{1+\sigma_i}\mathbb{1}[t = T_i] + \mathbb{1}[t > T_i], \tag{17}$$

where $T_i = 1/(\sigma_i\lambda_i)$. Thus, the i-th component is learned at time inversely proportional to $\lambda_i \cdot \sigma_i$.

Proof. The GF updates are given by:

$$\frac{\mathrm{d}\boldsymbol{W}(t)}{\mathrm{d}t} = -\left(\boldsymbol{Z}\boldsymbol{\Omega}^{1/2} - \boldsymbol{W}(t)\boldsymbol{H}(t)\boldsymbol{\Omega}^{1/2}\right)\boldsymbol{\Omega}^{1/2}\boldsymbol{H}(t)^\top = -\left(\boldsymbol{Z} - \boldsymbol{W}(t)\boldsymbol{H}(t)\right)\boldsymbol{\Omega}\boldsymbol{H}(t)^\top \tag{18a}$$

$$\frac{\mathrm{d}\boldsymbol{H}(t)}{\mathrm{d}t} = -\boldsymbol{W}(t)^\top\left(\boldsymbol{Z}\boldsymbol{\Omega}^{1/2} - \boldsymbol{W}(t)\boldsymbol{H}(t)\boldsymbol{\Omega}^{1/2}\right)\boldsymbol{\Omega}^{1/2} = -\boldsymbol{W}(t)^\top\left(\boldsymbol{Z} - \boldsymbol{W}(t)\boldsymbol{H}(t)\right)\boldsymbol{\Omega} \tag{18b}$$

where $\boldsymbol{W}(0) = e^{-\delta}\boldsymbol{U}\mathbf{R}^\top$ and $\boldsymbol{H}(0) = e^{-\delta}\mathbf{R}\boldsymbol{V}^\top$. As in (Saxe et al., 2013; 2019) change variables to $\overline{\boldsymbol{W}}(t), \overline{\boldsymbol{H}}(t)$ defined as

$$\boldsymbol{W}(t) = \boldsymbol{U}\overline{\boldsymbol{W}}(t)\mathbf{R}^\top, \qquad \text{and} \qquad \boldsymbol{H}(t) = \mathbf{R}\overline{\boldsymbol{H}}(t)\boldsymbol{V}^\top.$$

At $t = 0$, these matrices are diagonal and equal to $e^{-\delta}\mathbb{I}$ by initialization assumption. Substituting these in the update equations (18) and also using $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, we get that

$$\boldsymbol{U}\frac{\mathrm{d}\overline{\boldsymbol{W}}(t)}{\mathrm{d}t}\mathbf{R}^\top = \boldsymbol{U}\left(\boldsymbol{\Sigma} - \overline{\boldsymbol{W}}(t)\overline{\boldsymbol{H}}(t)\right)\boldsymbol{V}^\top\boldsymbol{\Omega}\boldsymbol{V}\overline{\boldsymbol{H}}(t)^\top\mathbf{R}^\top \tag{19a}$$

$$\mathbf{R}\frac{\mathrm{d}\overline{\boldsymbol{H}}(t)}{\mathrm{d}t}\boldsymbol{V}^\top = \mathbf{R}\overline{\boldsymbol{W}}(t)^\top\left(\boldsymbol{\Sigma} - \overline{\boldsymbol{W}}(t)\overline{\boldsymbol{H}}(t)\right)\boldsymbol{V}^\top\boldsymbol{\Omega}. \tag{19b}$$

Left and Right multiplying these with the partial unitary matrices $\boldsymbol{U}, \mathbf{R}, \boldsymbol{V}$ and recalling from (13) that $\boldsymbol{V}^\top \boldsymbol{\Omega} \boldsymbol{V} = \boldsymbol{\Lambda}$ (given in (14)), we arrive at

$$\frac{\mathrm{d}\overline{\boldsymbol{W}}(t)}{\mathrm{d}t} = -\left(\boldsymbol{\Sigma} - \overline{\boldsymbol{W}}(t)\overline{\boldsymbol{H}}(t)\right)\boldsymbol{\Lambda}\overline{\boldsymbol{H}}(t)^\top \tag{20a}$$

$$\frac{\mathrm{d}\overline{\boldsymbol{H}}(t)}{\mathrm{d}t} = -\overline{\boldsymbol{W}}(t)^\top\left(\boldsymbol{\Sigma} - \overline{\boldsymbol{W}}(t)\overline{\boldsymbol{H}}(t)\right)\boldsymbol{\Lambda}\,. \tag{20b}$$

At this point, recall that $\boldsymbol{\Lambda}$ is diagonal, and of course so is $\boldsymbol{\Sigma}$. Since $\overline{\boldsymbol{W}}, \overline{\boldsymbol{H}}$ are initialized to be diagonal matrices, they will remain diagonal throughout training. We thus arrive at a decoupled system of differential equations as in the unweighted case treated by (Saxe et al., 2013; Gidel et al., 2019).

In fact, the update equations are same as in the unweighted case (compare to (Gidel et al., 2019, Eq. 39)) with the only difference being the presence of the *effective weight matrix* $\boldsymbol{\Lambda}$. Having reached this point, we can follow almost the exact same steps as in the proof of (Gidel et al., 2019, Thm. 1). For brevity and because the adjustments are easy to check we directly present the formula for the diagonal entries $\overline{w}_i(t), \overline{h}_i(t)$ of $\overline{\boldsymbol{W}}(t), \overline{\boldsymbol{H}}(t)$ that solve (20) (see (Gidel et al., 2019, Eq. 48)). For $i \in [V-1]$:

$$\overline{w}_i(t) = \overline{h}_i(t) = \sqrt{\frac{\sigma_i e^{2\sigma_i\lambda_i t}}{\sigma_i e^{2\delta} - 1 + e^{2\sigma_i\lambda_i t}}} = \sqrt{\sigma_i}\sqrt{\frac{1}{1 + (\sigma_i e^{2\delta} - 1)e^{-2\sigma_i\lambda_i t}}} =: \sqrt{\sigma_i}\sqrt{\beta_i(t)}\,, \tag{21}$$

where we defined

$$\beta_i(t) := \frac{1}{1 + (\sigma_i e^{2\delta} - 1)e^{-2\sigma_i\lambda_i t}}\,, i \in [V-1]$$

Note the similarity to the unweighted case (Eq. (16)) only here the rate at which $\beta_i(t)$ approaches 1 is determined by $\sigma_i\lambda_i$, rather than by σ_i alone.

From this, we can also easily determine the limit of $\beta_i(\delta t)$ as $\delta \to \infty$ (see (Gidel et al., 2019, Eq. 64)):

$$\beta_i(\delta t) := \frac{e^{2\sigma_i\lambda_i\delta t}}{e^{2\sigma_i\lambda_i\delta t} + (\sigma_i e^{2\delta} - 1)} = \frac{e^{2\delta(\sigma_i\lambda_i t - 1)}}{e^{2\delta(\sigma_i\lambda_i t - 1)} + (\sigma_i - e^{-2\delta})} \overset{\delta\to\infty}{\longrightarrow} \frac{1}{1+\sigma_i}\mathbb{1}[t = \frac{1}{\sigma_i\lambda_i}] + \mathbb{1}[t > \frac{1}{\sigma_i\lambda_i}]\,. \tag{22}$$

□

Combining Proposition C.1 (for vanilla ERM) with Theorem C.2 (for reweighting) we arrive at the following corollary comparing the relative rates of learning of **`maj-maj`**, **`maj-min`**, **`min-min`** features.

Corollary C.3. *Assume the spectral initialization of Prop. C.1 and Thm. C.2 with vanishing initialization. Both vanillaERM and reweighting dynamics show a phase transition. On the one hand, Vanilla ERM learns the features in the order* **`maj-maj`**, **`maj-min`**, **`min-min`** *at times* $1/\sqrt{R}$, $1/\sqrt{(R+1)/2}$ *and* 1, *respectively. On the other hand, weigthed ERM (9) learns simultaneously the features* **`maj-maj`** *and* **`min-min`** *at time* $1/\sqrt{R+1}$ *before they learn the* **`maj-min`** *feature, closely after, at time* $\sqrt{2}/(\sqrt{R}+1)$.

Proof. This is direct consequence of the phase-transition like rates of Prop. C.1 and Thm C.2 at infinitesimal initialization. For reweighted loss, we further compute the products $\lambda_i\sigma_i$ of the effective weights λ_i in Eq. (14) and the diagonal entries σ_i of $\boldsymbol{\Sigma}$ in (11). □

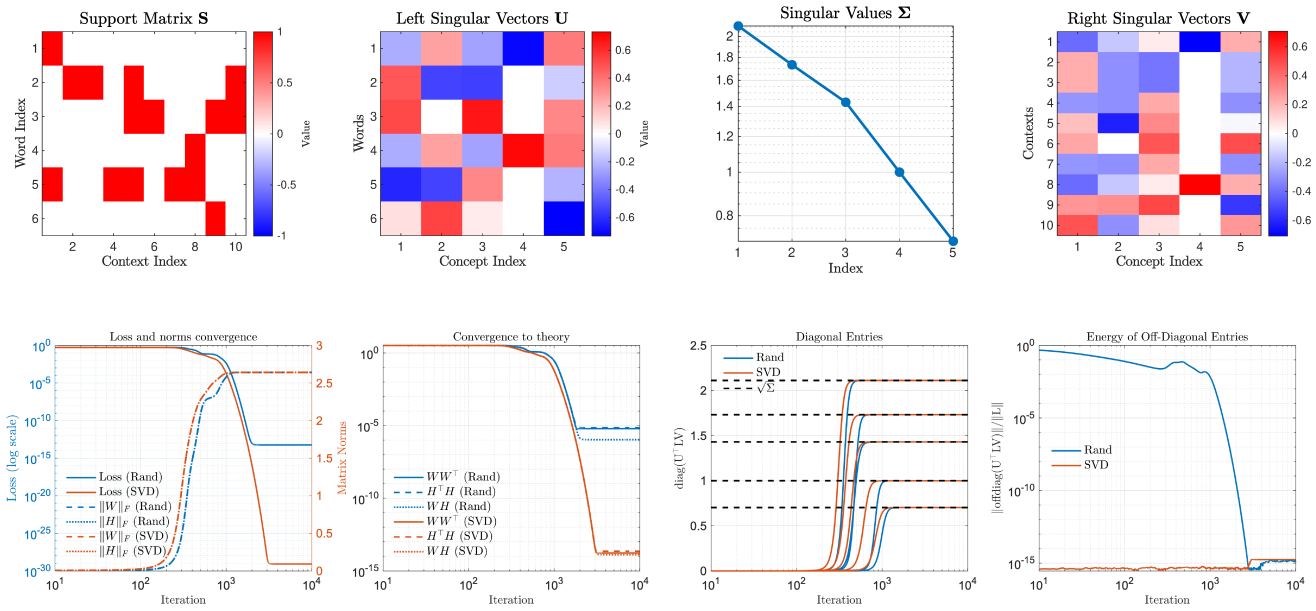

Figure 6. **(Top)** Support matrix and SVD factors of centered support matrix for a synthetic example. **(Bottom)** Training dynamics of GD minimization of UFM with square loss (Eq. (4) for two initializations: (i) SVD: initialize $\boldsymbol{W}$ and $\boldsymbol{H}$ as per Thm. C.1 for $\delta = 8$. (ii) Rand: intialize $\boldsymbol{W}$ and $\boldsymbol{H}$ random Gaussian scaled to match the norm of SVD initialization. Dynamics with the two initialization are shown in orange (SVD) and blue (Rand), respectively. Qualitatively the behavior is similar. *Left:* Training loss and norms of paramteres. *Middle-Left:* Convergence of word and context gram-matrices and of logits to the theory predicted by Thm. C.1. *Middle-Right:* Convergence of singular values of logit matrix to those of $\boldsymbol{\Sigma}$ (see Thm. C.1. *Right:* Projection of logits to subspace orthogonal to $\boldsymbol{U}$ and $\boldsymbol{V}$; Logits with Rand initialization initially have non-zero projection but it becomes zero as training progresses.