Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training

Cong Fang[a,1], Hangfeng He[a], Qi Long[b], and Weijie J. Su[c,2]

[a]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104; [b]Department of Biostatistics, Epidemiology, and Informatics, University of Pennsylvania, Philadelphia, PA 19104; and [c]Department of Statistics and Data Science, University of Pennsylvania, Philadelphia, PA 19104

Edited by David L. Donoho, Stanford University, Stanford, CA, and approved August 30, 2021 (received for review February 15, 2021)

In this paper, we introduce the *Layer-Peeled Model*, a nonconvex, yet analytically tractable, optimization program, in a quest to better understand deep neural networks that are trained for a sufficiently long time. As the name suggests, this model is derived by isolating the topmost layer from the remainder of the neural network, followed by imposing certain constraints separately on the two parts of the network. We demonstrate that the Layer-Peeled Model, albeit simple, inherits many characteristics of well-trained neural networks, thereby offering an effective tool for explaining and predicting common empirical patterns of deep-learning training. First, when working on class-balanced datasets, we prove that any solution to this model forms a simplex equiangular tight frame, which, in part, explains the recently discovered phenomenon of neural collapse [V. Papyan, X. Y. Han, D. L. Donoho, *Proc. Natl. Acad. Sci. U.S.A.* 117, 24652–24663 (2020)]. More importantly, when moving to the imbalanced case, our analysis of the Layer-Peeled Model reveals a hitherto-unknown phenomenon that we term *Minority Collapse*, which fundamentally limits the performance of deep-learning models on the minority classes. In addition, we use the Layer-Peeled Model to gain insights into how to mitigate Minority Collapse. Interestingly, this phenomenon is first predicted by the Layer-Peeled Model before being confirmed by our computational experiments.

deep learning | neural collapse | class imbalance

Introduction

In the past decade, deep learning has achieved remarkable performance across a range of scientific and engineering domains (1–3). Interestingly, these impressive accomplishments were mostly achieved by heuristics and tricks, though often plausible, without much principled guidance from a theoretical perspective. On the flip side, however, this reality suggests the great potential a theory could have for advancing the development of deep-learning methodologies in the coming decade.

Unfortunately, it is not easy to develop a theoretical foundation for deep learning. Perhaps the most difficult hurdle lies in the nonconvexity of the optimization problem for training neural networks, which, loosely speaking, stems from the interaction between different layers of neural networks. To be more precise, consider a neural network for K-class classification (in logits), which in its simplest form reads*

$$\boldsymbol{f}(\boldsymbol{x};\boldsymbol{W}_{\text{full}}) = \boldsymbol{b}_L + \boldsymbol{W}_L\sigma(\boldsymbol{b}_{L-1} + \boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{b}_1 + \boldsymbol{W}_1\boldsymbol{x})\cdots)).$$

Here, $\boldsymbol{W}_{\text{full}} := \{\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_L\}$ denotes the weights of the L layers, $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_L\}$ denotes the biases, and $\sigma(\cdot)$ is a nonlinear activation function such as the rectified linear unit (ReLU). Owing to the complex and nonlinear interaction between the L layers, when applying stochastic gradient descent to the optimization problem

$$\min_{\boldsymbol{W}_{\text{full}}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}_{k,i};\boldsymbol{W}_{\text{full}}),\boldsymbol{y}_k) + \frac{\lambda}{2}\|\boldsymbol{W}_{\text{full}}\|^2, \quad [1]$$

with a loss function $\mathcal{L}$ for training the neural network, it becomes very difficult to pinpoint how a given layer influences the output $\boldsymbol{f}$ (above, $\{\boldsymbol{x}_{k,i}\}_{i=1}^{n_k}$ denotes the training examples in the k-th class, with label $\boldsymbol{y}_k$, $N = n_1 + \cdots + n_K$ is the total number of training examples, $\lambda > 0$ is the weight decay parameter, and $\|\cdot\|$ throughout the paper is the ℓ_2 norm). Worse, this difficulty in analyzing deep-learning models is compounded by an ever-growing number of layers.

Therefore, any attempt to develop a tractable and comprehensive theory for demystifying deep learning would presumably first need to simplify the interaction between a large number of layers. Following this intuition, in this paper, we introduce the following optimization program as a *surrogate* model for Eq. **1** with the goal of unveiling quantitative patterns of deep neural networks:

$$\min_{\boldsymbol{W}_L,\boldsymbol{H}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}_L\boldsymbol{h}_{k,i},\boldsymbol{y}_k)$$
$$\text{s.t. } \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W, \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H, \quad [2]$$

Significance

The remarkable development of deep learning over the past decade relies heavily on sophisticated heuristics and tricks. To better exploit its potential in the coming decade, perhaps a rigorous framework for reasoning about deep learning is needed, which, however, is not easy to build due to the intricate details of neural networks. For near-term purposes, a practical alternative is to develop a mathematically tractable surrogate model, yet maintaining many characteristics of neural networks. This paper proposes a model of this kind that we term the Layer-Peeled Model. The effectiveness of this model is evidenced by, among others, its ability to reproduce a known empirical pattern and to predict a hitherto-unknown phenomenon when training deep-learning models on imbalanced datasets.

Author contributions: C.F., H.H., Q.L., and W.J.S. designed research; C.F., H.H., and W.J.S. performed research; C.F., H.H., and W.J.S. contributed new reagents/analytic tools; C.F., H.H., and W.J.S. analyzed data; and C.F., H.H., Q.L., and W.J.S. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]Present address: Department of Key Laboratory of Machine Perception, Peking University, Beijing 100871, China.

[2]To whom correspondence may be addressed. Email: suw@wharton.upenn.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2103091118/-/DCSupplemental.

Published October 20, 2021.

*The softmax step is implicitly included in the loss function, and we omit other operations such as max-pooling for simplicity.

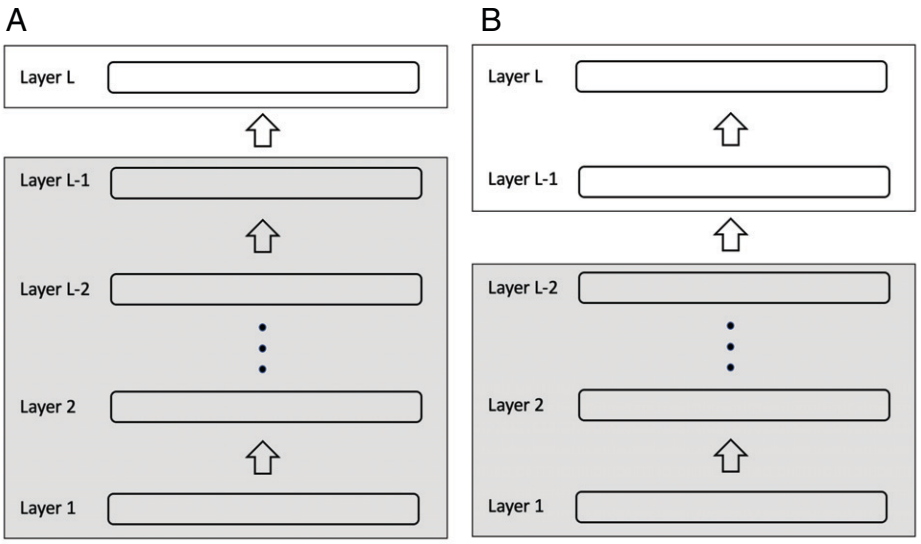

Fig. 1. Illustration of Layer-Peeled Models. *B* represents the 2-Layer-Peeled Model, which is discussed in Section 6. For each panel, we preserve the details of the white (top) box, whereas the gray (bottom) box is modeled by a simple decision variable for every training example. (*A*) The 1-Layer-Peeled Model. (*B*) The 2-Layer-Peeled Model.

where $\boldsymbol{W}_L = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]^\top \in \mathbb{R}^{K\times p}$ is, as in Eq. **1**, comprised of K linear classifiers in the last layer, $\boldsymbol{H} = [\boldsymbol{h}_{k,i} : 1 \le k \le K, 1 \le i \le n_k] \in \mathbb{R}^{p\times N}$ corresponds to the p-dimensional last-layer activations/features of all N training examples, and E_H and E_W are two positive scalars. Note that the bias terms are omitted for simplicity. Although still nonconvex, this optimization program is presumably much more amenable to analysis than the old one, Eq. **1**, as the interaction now is only between two layers.

In relating Eq. **2** to Eq. **1**, a first simple observation is that $\boldsymbol{f}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{\text{full}}) = \boldsymbol{W}_L\sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}_{k,i})\cdots))$ in Eq. **1** is replaced by $\boldsymbol{W}_L\boldsymbol{h}_{k,i}$ in Eq. **2**. Put differently, the black-box nature of the last-layer features, namely, $\sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}_{k,i})\cdots))$ is now modeled by a simple decision variable $\boldsymbol{h}_{k,i}$ for each training example, with an overall constraint on their ℓ_2 norm. Intuitively speaking, this simplification is done by *peeling* off the topmost layer from the neural network. Thus, we call the optimization program (1) the *1-Layer-Peeled Model*, or simply the *Layer-Peeled Model*.

At a high level, the Layer-Peeled Model takes a *top-down* approach to the analysis of deep neural networks. As illustrated in Fig. 1, the essence of the modeling strategy is to break down the neural network from top to bottom, specifically singling out the topmost layer and modeling all bottom layers collectively as a single variable. In fact, the top-down perspective that we took in the development of the Layer-Peeled Model was inspired by a recent breakthrough made by Papyan, Han, and Donoho (4), who discovered a mathematically elegant and pervasive phenomenon termed neural collapse in deep-learning training. This top-down approach was also taken in refs. (5–9) to investigate various aspects of deep-learning models.

Two Applications. Despite its plausibility, the ultimate test of the Layer-Peeled Model lies in its ability to faithfully approximate deep-learning models through explaining empirical observations and, better, predicting new phenomena. In what follows, we provide convincing evidence that the Layer-Peeled Model is up to this task by presenting two findings. To be concrete, we remark that the results below are concerned with well-trained deep-learning models, which correspond to, in rough terms, (near) optimal solutions of Eq. **1**.

Balanced data. Roughly speaking, neural collapse (4) refers to the emergence of certain geometric patterns of the last-layer features $\sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}_{k,i})\cdots))$ and the last-layer classifiers $\boldsymbol{W}_L$, when the neural network for *balanced* classification problems is well-trained in the sense that it is toward not only zero misclassification error, but also negligible[†] cross-entropy loss. Specifically, the authors observed the following properties in their massive experiments: The last-layer features from the same class tend to be very close to their class mean; these K-class means centered at the global mean have the same length and form the maximally possible equal-sized angles between any pair; moreover, the last-layer classifiers become dual to the class means in the sense that they are equal to each other for each class up to a scaling factor. See a more precise description in Section B.

While it seems hopeless to rigorously prove neural collapse for multiple-layer neural networks (Eq. **1**) at the moment, alternatively, we seek to show that this phenomenon emerges in the surrogate model (Eq. **2**). More precisely, when the size of each class $n_k = n$ for all k, is it true that any global minimizer $\boldsymbol{W}_L^\star = [\boldsymbol{w}_1^\star, \ldots, \boldsymbol{w}_K^\star]^\top$, $\boldsymbol{H}^\star = [\boldsymbol{h}_{k,i}^\star : 1 \le k \le K, 1 \le i \le n]$ of Eq. **2** exhibits neural collapse? The following result answers this question in the affirmative:

Finding 1. *Neural collapse occurs in the Layer-Peeled Model.*

A formal statement of this result and a detailed discussion are given in Section 3.

This result applies to a family of loss functions $\mathcal{L}$, particularly including the cross-entropy loss and the contrastive loss (see, e.g., ref. (10)). As an immediate implication, this result provides evidence of the Layer-Peeled Model's ability to characterize well-trained deep-learning models.

Imbalanced data. While a surrogate model would be satisfactory if it explains some already-observed phenomenon, we set a *higher* standard for the model, asking whether it can predict a *new* common empirical pattern. Encouragingly, the Layer-Peeled Model happens to meet this standard. Specifically, we consider training deep-learning models on imbalanced datasets, where some classes contain many more training examples than others. Despite the pervasiveness of imbalanced classification in many

[†]Strictly speaking, in the presence of an ℓ_2 regularization term, which is equivalent to weight decay, the cross-entropy loss evaluated at any global minimizer of Eq. **1**. is bounded away from 0.

practical applications (11), the literature remains scarce on its impact on the trained neural networks from a theoretical standpoint. Here, we provide mathematical insights into this problem by using the Layer-Peeled Model. In the following result, we consider optimal solutions to the Layer-Peeled Model on a dataset with two different class sizes: The first K_A majority classes each contain n_A training examples ($n_1 = n_2 = \cdots = n_{K_A} = n_A$), and the remaining $K_B := K - K_A$ minority classes each contain n_B examples ($n_{K_A+1} = n_{K_A+2} = \cdots = n_K = n_B$). We call $R := n_A/n_B > 1$ the imbalance ratio.

Finding 2. *In the Layer-Peeled Model, the last-layer classifiers corresponding to the minority classes, namely, $\boldsymbol{w}^\star_{K_A+1}, \boldsymbol{w}^\star_{K_A+2}, \ldots, \boldsymbol{w}^\star_K$, collapse to a single vector when R is sufficiently large.*

This result is elaborated on in Section 4. The derivation involves some elements to tackle the nonconvexity of the Layer-Peeled Model (Eq. **2**) and the asymmetry due to the imbalance in class sizes.

In slightly more detail, we identify a phase transition as the imbalance ratio R increases: When R is below a threshold, the minority classes are distinguishable in terms of their last-layer classifiers; when R is above the threshold, they become indistinguishable. While this phenomenon is merely predicted by the simple Layer-Peeled Model (Eq. **2**), it appears in our computational experiments on deep neural networks. More surprisingly, our prediction of the phase transition point is in excellent agreement with the experiments, as shown in Fig. 2.

This phenomenon, which we refer to as *Minority Collapse*, reveals the fundamental difficulty in using deep learning for classification when the dataset is widely imbalanced, even in terms of optimization, not to mention generalization. This is not a priori evident given that neural networks have a large approximation capacity (see, e.g., ref. (14)). Importantly, Minority Collapse emerges at a finite value of the imbalance ratio rather than at infinity. Moreover, even below the phase transition point of this ratio, we find that the angles between any pair of the minority classifiers are already smaller than those of the majority classes, both theoretically and empirically.

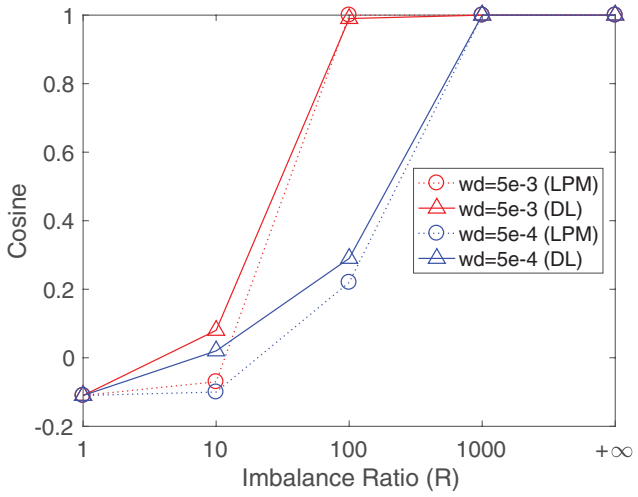

Fig. 2. Minority Collapse predicted by the Layer-Peeled Model (LPM; in dotted lines) and empirically observed in deep learning (DL; in solid lines) on imbalanced datasets with $K_A = 7$ and $K_B = 3$. The y axis denotes the average cosine of the angles between any pair of the minority classifier $w^\star_{K_A+1}, \ldots, w^\star_K$ for both LPM and DL. The datasets we use are subsets of the CIFAR10 datasets (12), and the size of the majority classes is fixed to 5,000. The experiments use VGG13 (13) as the deep-learning architecture, with weight decay (wd) $\lambda = 5 \times 10^{-3}, 5 \times 10^{-4}$. The prediction is especially accurate in capturing the phase transition point where the cosine becomes 1 or, equivalently, the minority classifiers become parallel to each other. More details can be found in Section C.

Related Work. There is a venerable line of work attempting to gain insights into deep learning from a theoretical point of view (15–29). See also the reviews (30–33) and references therein.

The work of neural collapse by ref. (4) in this body of work is particularly noticeable with its mathematically elegant and convincing insights. In brief, ref. (4) observed the following four properties of the last-layer features and classifiers in deep-learning training on balanced datasets:[‡]

(NC1) Variability collapse: The within-class variation of the last-layer features becomes 0, which means that these features collapse to their class means.
(NC2) The class means centered at their global mean collapse to the vertices of a simplex equiangular tight frame (ETF) up to scaling.
(NC3) Up to scaling, the last-layer classifiers each collapse to the corresponding class means.
(NC4) The network's decision collapses to simply choosing the class with the closest Euclidean distance between its class mean and the activations of the test example.

Now we give the formal definition of ETF (4, 34).

Definition 1. A K-simplex ETF is a collection of points in $\mathbb{R}^p$ specified by the columns of the matrix

$$\boldsymbol{M}^\star = \sqrt{\frac{K}{K-1}}\boldsymbol{P}\left(\boldsymbol{I}_K - \frac{1}{K}\boldsymbol{1}_K\boldsymbol{1}_K^\top\right),$$

where $\boldsymbol{I}_K \in \mathbb{R}^{K\times K}$ is the identity matrix, $\boldsymbol{1}_K$ is the ones vector, and $\boldsymbol{P} \in \mathbb{R}^{p\times K}$ $(p \geq K)$[§] is a partial orthogonal matrix such that $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}_K$.

A common setup of the experiments for validating neural collapse is the use of the cross-entropy loss with ℓ_2 regularization, which corresponds to weight decay in stochastic gradient descent. Based on convincing arguments and numerical evidence, ref. (4) demonstrated that the symmetry and stability of neural collapse improve deep-learning training in terms of generalization, robustness, and interpretability. Notably, these improvements occur with the benign overfitting phenomenon (35–39) during the terminal phase of training—when the trained model interpolates the in-sample training data.

In passing, we remark that concurrent works (40–43) produced neural collapse using different surrogate models. In slightly more detail, refs. (40–42) obtained their models by peeling off the topmost layer. The difference, however, is that refs. (41) and (42) considered models that impose a norm constraint for each class, as opposed to an overall constraint, as employed in the Layer-Peeled Model. Moreover, ref. (40) analyzed gradient flow with an unconstrained features model using the squared loss instead of the cross-entropy loss. The work in ref. (43) provided an insightful perspective for the analysis of neural networks using convex duality. Relying on a convex formulation that is in the same spirit as our semidefinite programming relaxation, the authors of ref. (43) observed neural collapse in their ReLU-based model by leveraging strong duality under certain conditions.

Derivation

In this section, we heuristically derive the Layer-Peeled Model as an analytical surrogate for well-trained neural networks. Although our derivation lacks rigor, the goal is to reduce the complexity of the optimization problem (Eq. **1**) while roughly

[‡]See the mathematical description of neural collapse in Theorem 1.

[§]To be complete, we only require $p \geq K - 1$. When $p = K - 1$, we can choose P such that $\left[\boldsymbol{P}^\top, \boldsymbol{1}_K\right]$ is an orthogonal matrix.

preserving its structure. Notably, the penalty $\frac{\lambda}{2}\|\boldsymbol{W}_{\text{full}}\|^2$ corresponds to weight decay used in training deep-learning models, which is necessary for preventing this optimization program from attaining its minimum at infinity when $\mathcal{L}$ is the cross-entropy loss. For simplicity, we omit the biases in the neural network $\boldsymbol{f}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{\text{full}})$.

Taking a top-down standpoint, our modeling strategy starts by singling out the weights $\boldsymbol{W}_L$ of the topmost layer and rewriting Eq. **1** as

$$\min_{\boldsymbol{W}_L, \boldsymbol{H}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}_L\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}), \boldsymbol{y}_k) + \frac{\lambda}{2}\|\boldsymbol{W}_L\|^2 + \frac{\lambda}{2}\|\boldsymbol{W}_{-L}\|^2, \quad [3]$$

where the last-layer feature function $\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}) := \sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}_{k,i})\cdots))$ and $\boldsymbol{W}_{-L}$ denotes the weights from all layers but the last layer. From the Lagrangian dual viewpoint, a minimum of the optimization program above is also an optimal solution to

$$\min_{\boldsymbol{W}_L, \boldsymbol{W}_{-L}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}_L\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}), \boldsymbol{y}_k) \quad \text{s.t.} \quad \|\boldsymbol{W}_L\|^2 \le C_1, \|\boldsymbol{W}_{-L}\|^2 \le C_2, \quad [4]$$

for some positive numbers C_1 and C_2.[¶] To clear up any confusion, note that due to its nonconvexity, Eq. **3** may admit multiple global minima, and each in general corresponds to different values of C_1, C_2. Next, we can equivalently write Eq. **4** as

$$\min_{\boldsymbol{W}_L, \boldsymbol{H}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}_L\boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \quad \text{s.t.} \quad \|\boldsymbol{W}_L\|^2 \le C_1, \quad \boldsymbol{H} \in \left\{\boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \le C_2\right\}, \quad [5]$$

where $\boldsymbol{H} = [\boldsymbol{h}_{k,i} : 1 \le k \le K, 1 \le i \le n_k]$ denotes a decision variable, and the function $\boldsymbol{H}(\boldsymbol{W}_{-L})$ is defined as $\boldsymbol{H}(\boldsymbol{W}_{-L}) := [\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}) : 1 \le k \le K, 1 \le i \le n_k]$ for any $\boldsymbol{W}_{-L}$.

To simplify Eq. **5**, we make the *ansatz* that the range of $\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L})$ under the constraint $\|\boldsymbol{W}_{-L}\|^2 \le C_2$ is approximately an ellipse in the sense that

$$\left\{\boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \le C_2\right\} \approx \left\{\boldsymbol{H} : \sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le C_2'\right\}, \quad [6]$$

for some $C_2' > 0$. Loosely speaking, this ansatz asserts that $\boldsymbol{H}$ should be regarded as a variable in an ℓ_2 space. To shed light on the rationale behind the ansatz, note that $\boldsymbol{h}_{k,i}$ intuitively lives in the dual space of $\boldsymbol{W}$ in view of the appearance of the product $\boldsymbol{W}\boldsymbol{h}_{k,i}$ in the objective. Furthermore, $\boldsymbol{W}$ is in an ℓ_2 space for the ℓ_2 constraint on it. Last, note that ℓ_2 spaces are self-dual.

[¶] Denoting by $(\boldsymbol{W}_L^\star, \boldsymbol{W}_{-L}^\star)$ an optimal solution to Eq. **3**, then we can take $C_1 = \|\boldsymbol{W}_L^\star\|^2$ and $C_2 = \|\boldsymbol{W}_{-L}^\star\|^2$.

Inserting this approximation into Eq. **5**, we obtain the following optimization program, which we call the Layer-Peeled Model:

$$\min_{\boldsymbol{W}, \boldsymbol{H}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \quad \text{s.t.} \quad \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W, \quad \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H. \quad [7]$$

For simplicity, above and henceforth we write $\boldsymbol{W} := \boldsymbol{W}_L \equiv [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]^\top$ for the last-layer classifiers/weights and the thresholds $E_W = C_1/K$ and $E_H = C_2'/K$.

This optimization program is nonconvex but, as we will show soon, is generally mathematically tractable for analysis. On the surface, the Layer-Peeled Model has no dependence on the data $\{\boldsymbol{x}_{k,i}\}$, which, however, is not the correct picture, since the dependence has been implicitly incorporated into the threshold E_H.

In passing, we remark that neural collapse does *not* emerge if the second constraint of Eq. **7** uses the ℓ_q norm for any $q \ne 2$ (strictly speaking, ℓ_q is not a norm when $q < 1$), in place of the ℓ_2 norm. This fact in turn justifies in part the ansatz Eq. **6**. This result is formally stated in Proposition 2 in Section 6.

Layer-Peeled Model for Explaining Neural Collapse

In this section, we consider training deep neural networks on a balanced dataset—that is, $n_k = n$ for all classes $1 \le k \le K$. Our main finding is that the Layer-Peeled Model displays the neural collapse phenomenon, just as in deep-learning training (4). The proofs are all deferred to *SI Appendix*. Throughout this section, we assume $p \ge K - 1$ unless otherwise specified. This assumption is satisfied in many popular network architectures, where p is usually tens or hundreds of times of K.

Cross-Entropy Loss. The cross-entropy loss is perhaps the most popular loss used in training deep-learning models for classification tasks. This loss function takes the form

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{y}_k) = -\log\left(\frac{\exp(\boldsymbol{z}(k))}{\sum_{k'=1}^{K}\exp(\boldsymbol{z}(k'))}\right), \quad [8]$$

where $\boldsymbol{z}(k')$ denotes the k'-th entry of the logit $\boldsymbol{z}$. Recall that $\boldsymbol{y}_k$ is the label of the k-th class, and the feature $\boldsymbol{z}$ is set to $\boldsymbol{W}\boldsymbol{h}_{k,i}$ in the Layer-Peeled Model (Eq. **7**). In contrast to the complex deep neural networks, which are often considered a black-box, the Layer-Peeled Model is much easier to deal with. As an exemplary use case, the following result shows that any minimizer of the Layer-Peeled Model (Eq. **7**) with the cross-entropy loss admits an almost closed-form expression.

Theorem 1. *In the balanced case, any global minimizer* $\boldsymbol{W}^\star \equiv [\boldsymbol{w}_1^\star, \ldots, \boldsymbol{w}_K^\star]^\top$, $\boldsymbol{H}^\star \equiv [\boldsymbol{h}_{k,i}^\star : 1 \le k \le K, 1 \le i \le n]$ *of Eq.* **7** *with the cross-entropy loss obeys*

$$\boldsymbol{h}_{k,i}^\star = C\boldsymbol{w}_k^\star = C'\boldsymbol{m}_k^\star, \quad [9]$$

for all $1 \le i \le n, 1 \le k \le K$, *where the constants* $C = \sqrt{E_H/E_W}$, $C' = \sqrt{E_H}$, *and the matrix* $[\boldsymbol{m}_1^\star, \ldots, \boldsymbol{m}_K^\star]$ *forms a K-simplex ETF specified in Definition* 1.

***Remark* 2.** Note that the minimizers $(\boldsymbol{W}^\star, \boldsymbol{H}^\star)$'s are equivalent to each other up to rotation. This is because of the rational invariance of simplex ETFs (see $\boldsymbol{P}$ in Definition 1).

This theorem demonstrates the highly symmetric geometry of the last-layer features and weights of the Layer-Peeled Model, which is precisely the phenomenon of neural collapse. Explicitly,

Eq. **9** says that all within-class (last-layer) features are the same: $\boldsymbol{h}^\star_{k,i} = \boldsymbol{h}^\star_{k,i'}$ for all $1 \le i, i' \le n$; next, it also says that the K-class-mean features $\boldsymbol{h}^\star_k := \boldsymbol{h}^\star_{k,i}$ together exhibit a K-simplex ETF up to scaling, from which we immediately conclude that

$$\cos \measuredangle(\boldsymbol{h}^\star_k, \boldsymbol{h}^\star_{k'}) = -\frac{1}{K-1}, \qquad \textbf{[10]}$$

for any $k \neq k'$ by Definition 1;[‖] in addition, Eq. **9** also displays the precise duality between the last-layer classifiers and features. Taken together, these facts indicate that the minimizer $(\boldsymbol{W}^\star, \boldsymbol{H}^\star)$ satisfies exactly (NC1)–(NC3). Last, Property (NC4) is also satisfied by recognizing that, for any given last-layer features $\boldsymbol{h}$, the predicted class is $\arg\max_k \boldsymbol{w}^\star_k \cdot \boldsymbol{h}$, where $\boldsymbol{a} \cdot \boldsymbol{b}$ denotes the inner product of the two vectors. Note that the prediction satisfies

$$\arg\max_k \boldsymbol{w}^\star_k \cdot \boldsymbol{h} = \arg\max_k \boldsymbol{h}^\star_k \cdot \boldsymbol{h} = \arg\min_k \|\boldsymbol{h}^\star_k - \boldsymbol{h}\|^2.$$

Conversely, the presence of neural collapse in the Layer-Peeled Model offers evidence of the effectiveness of our model as a tool for analyzing neural networks. To be complete, we remark that other models were very recently proposed to justify the neural collapse phenomenon (40–42) (see also ref. (44)).

Extensions to Other Loss Functions. In the modern practice of deep learning, various loss functions are employed to take into account the problem characteristics. Here, we show that the Layer-Peeled Model continues to exhibit the phenomenon of neural collapse for some popular loss functions.

Contrastive loss. Contrastive losses have been extensively used recently in both supervised and unsupervised deep learning (10, 45–47). These losses pull similar training examples together in their embedding space while pushing apart dissimilar examples. Here, we consider the supervised contrastive loss (48), which (in the balanced case) is defined through the last-layer features by introducing $\mathcal{L}_c$ as

$$\frac{1}{n}\sum_{j=1}^{n} -\log\left(\frac{\exp(\boldsymbol{h}_{k,i}\cdot\boldsymbol{h}_{k,j}/\tau)}{\sum_{k'=1}^{K}\sum_{\ell=1}^{n}\exp(\boldsymbol{h}_{k,i}\cdot\boldsymbol{h}_{k',\ell}/\tau)}\right), \qquad \textbf{[11]}$$

where $\tau > 0$ is a parameter. Note that this loss function uses the label information implicitly. As the loss does not involve the last-layer classifiers explicitly, the Layer-Peeled Model in this case takes the form[**]

$$\begin{aligned} \min_{\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_c(\boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\ \text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{h}_{k,i}\|^2 \le E_H. \end{aligned} \qquad \textbf{[12]}$$

We show that this Layer-Peeled Model also exhibits neural collapse in its last-layer features, even though the label information is not explicitly explored in the loss.

Theorem 3. *Any global minimizer of Eq.* **12** *satisfies*

$$\boldsymbol{h}^\star_{k,i} = \sqrt{E_H}\,\boldsymbol{m}^\star_k, \qquad \textbf{[13]}$$

for all $1 \le k \le K$ *and* $1 \le i \le n$, *where* $[\boldsymbol{m}^\star_1, \ldots, \boldsymbol{m}^\star_K]$ *forms a K-simplex ETF.*

Theorem 3 shows that the contrastive loss in the associated Layer-Peeled Model does a perfect job in pulling together training examples from the same class. Moreover, as seen from the denominator in Eq. **11**, minimizing this loss would intuitively render the between-class inner products of last-layer features as small as possible, thereby pushing the features to form the vertices of a K-simplex ETF up to scaling.

Softmax-based loss. The cross-entropy loss can be thought of as a softmax-based loss. To see this, define the softmax transform as

$$\boldsymbol{S}(\boldsymbol{z}) = \left[\frac{\exp(\boldsymbol{z}((1)))}{\sum_{k=1}^{K}\exp(\boldsymbol{z}(k))}, \ldots, \frac{\exp(\boldsymbol{z}(K))}{\sum_{k=1}^{K}\exp(\boldsymbol{z}(k))}\right]^\top,$$

for $\boldsymbol{z} \in \mathbb{R}^K$. Let g_1 be any nonincreasing convex function and g_2 be any nondecreasing convex function, both defined on $(0, 1)$. We consider a softmax-based loss function that takes the form

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{y}_k) = g_1\left(\boldsymbol{S}(\boldsymbol{z})(k)\right) + \sum_{k'=1,\ k'\neq k}^{K} g_2\left(\boldsymbol{S}(\boldsymbol{z})(k')\right). \qquad \textbf{[14]}$$

Here, $\boldsymbol{S}(\boldsymbol{z})(k)$ denotes the k-th element of $\boldsymbol{S}(\boldsymbol{z})$. Taking $g_1(x) = -\log x$ and $g_2 \equiv 0$, we recover the cross-entropy loss. Another example is to take $g_1(x) = (1-x)^q$ and $g_2(x) = x^q$ for $q > 1$, which can be implemented in most deep-learning libraries, such as PyTorch (49).

We have the following theorem regarding the softmax-based loss functions in the balanced case.

Theorem 4. *Assume* $\sqrt{E_H E_W} > \frac{K-1}{K}\log\left(K^2\sqrt{E_H E_W} + (2K-1)(K-1)\right)$. *For any loss function defined in Eq.* **14**, $(\boldsymbol{W}^\star, \boldsymbol{H}^\star)$ *given by Eq.* **9** *is a global minimizer of Eq.* **7**. *Moreover, if* g_2 *is strictly convex and at least one of* g_1, g_2 *is strictly monotone, then any global minimizer must be given by Eq.* **9**.

In other words, neural collapse continues to emerge with softmax-based losses under mild regularity conditions. The first part of this theorem does not preclude the possibility that the Layer-Peeled Model admits solutions other than Eq. **9**. When applied to the cross-entropy loss, it is worth pointing out that this theorem is a weak version of Theorem 1, albeit more general. Regarding the first assumption in Theorem 4, note that E_H and E_W would be arbitrarily large if the weight decay λ in Eq. **1** is sufficiently small, thereby meeting the assumption concerning $\sqrt{E_H E_W}$ in this theorem.

We remark that Theorem 4 does not require the convexity of the loss $\mathcal{L}$. To circumvent the hurdle of nonconvexity, our proof in *SI Appendix* presents several elements.

In passing, we leave the experimental confirmation of neural collapse with these loss functions for future work.

Layer-Peeled Model for Predicting Minority Collapse

Deep-learning models are often trained on datasets where there is a disproportionate ratio of observations in each class (50–52). For example, in the Places2 challenge dataset (53), the number of images in its majority scene categories is about eight times that in its minority classes. Another example is the Ontonotes dataset for part-of-speech tagging (54), where the number of words in its majority classes can be more than 100 times that in its minority classes. While empirically, the imbalance in class sizes often leads to inferior model performance of deep learning (see, e.g., ref. (11)), there remains a lack of a solid theoretical footing for understanding its effect, perhaps due to the complex details of deep-learning training.

In this section, we use the Layer-Peeled Model to seek a fine-grained characterization of how class imbalance impacts neural networks that are trained for a sufficiently long time. In particular, neural collapse no longer emerges in the presence of class imbalance (see numerical evidence in *SI Appendix*, Fig. S2). Instead, our analysis predicts a phenomenon we term *Minority*

[‖] Note that the cosine value $-\frac{1}{K-1}$ corresponds to the largest possible angle for any K points that have an equal ℓ_2 norm and equal-sized angles between any pair. As pointed out in (4), the largest angle implies a large-margin solution (6).

[**] In Eq. **11**, $h_{k,i} \equiv h(x_{k,i}, \boldsymbol{W}_{-L})$ depends on the data, whereas in Eq. **12**, $h_{k,i}$'s form the decision variable $\boldsymbol{H}$.

Collapse, which fundamentally limits the performance of deep learning, especially on the minority classes, both theoretically and empirically. All omitted proofs are relegated to *SI Appendix*.

Technique: Convex Relaxation. When it comes to imbalanced datasets, the Layer-Peeled Model no longer admits a simple expression for its minimizers as in the balanced case, due to the lack of symmetry between classes. This fact results in, among others, an added burden on numerically computing the solutions of the Layer-Peeled Model.

To overcome this difficulty, we introduce a convex optimization program as a relaxation of the nonconvex Layer-Peeled Model (Eq. **7**), relying on the well-known result for relaxing a quadratically constrained quadratic program as a semidefinite program (see, e.g., ref. (55)). To begin with, defining $\boldsymbol{h}_k$ as the feature mean of the k-th class (i.e., $\boldsymbol{h}_k := \frac{1}{n_k}\sum_{i=1}^{n_k}\boldsymbol{h}_{k,i}$), we introduce a decision variable $\boldsymbol{X} := \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right]^\top \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right] \in \mathbb{R}^{2K\times 2K}$. By definition, $\boldsymbol{X}$ is positive semidefinite and satisfies

$$\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{X}(k,k) = \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{h}_k\|^2 \overset{a}{\leq} \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \leq E_H,$$

and

$$\frac{1}{K}\sum_{k=K+1}^{2K}\boldsymbol{X}(k,k) = \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \leq E_W,$$

where $\overset{a}{\leq}$ follows from the Cauchy–Schwarz inequality. Thus, we consider the following semidefinite programming problem:[#]

$$\begin{aligned}\min_{\boldsymbol{X}\in\mathbb{R}^{2K\times 2K}} \ & \sum_{k=1}^{K}\frac{n_k}{N}\mathcal{L}(\boldsymbol{z}_k, \boldsymbol{y}_k)\\ \text{s.t. } & \boldsymbol{X}\succeq 0,\\ & \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{X}(k,k)\leq E_H,\ \frac{1}{K}\sum_{k=K+1}^{2K}\boldsymbol{X}(k,k)\leq E_W,\\ & \text{for all} 1\leq k\leq K,\\ & \boldsymbol{z}_k = [\boldsymbol{X}(k,K+1), \boldsymbol{X}(k,K+2), \ldots, \boldsymbol{X}(k,2K)]^\top. \end{aligned} \quad \textbf{[15]}$$

Lemma 1 below relates the solutions of Eq. **15** to that of Eq. **7**.

Lemma 1. *Assume $p \geq 2K$ and the loss function $\mathcal{L}$ is convex in its first argument. Let $\boldsymbol{X}^\star$ be a minimizer of the convex program* **[15]**. *Define $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ as*

$$\begin{aligned}&\left[\boldsymbol{h}_1^\star, \boldsymbol{h}_2^\star, \ldots, \boldsymbol{h}_K^\star,\ (\boldsymbol{W}^\star)^\top\right] = \boldsymbol{P}(\boldsymbol{X}^\star)^{1/2},\\ &\boldsymbol{h}_{k,i}^\star = \boldsymbol{h}_k^\star,\ \text{for all} 1\leq i\leq n, 1\leq k\leq K,\end{aligned} \quad \textbf{[16]}$$

where $(\boldsymbol{X}^\star)^{1/2}$ denotes the positive square root of $\boldsymbol{X}^\star$ and $\boldsymbol{P}\in\mathbb{R}^{p\times 2K}$ is any partial orthogonal matrix such that $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}_{2K}$. Then, $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ is a minimizer of Eq. **7**. *Moreover, if all $\boldsymbol{X}^\star$'s satisfy $\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{X}^\star(k,k) = E_H$, then all the solutions of Eq.* **7**. *are in the form of Eq.* **16**.

This lemma in effect says that the relaxation does *not* lead to any loss of information when we study the Layer-Peeled Model through a convex program, thereby offering a computationally efficient tool for gaining insights into the terminal phase of training deep neural networks on imbalanced datasets. An appealing feature is that the size of the program **[15]** is independent of the number of training examples. Besides, this lemma predicts that even in the imbalanced case, the last-layer features collapse to their class means under mild conditions. Therefore, Property (NC1) is satisfied (see more discussion about the condition in *SI Appendix*).

The assumption of the convexity of $\mathcal{L}$ in the first argument is satisfied by a large class of loss functions. The condition that the first K-diagonal elements of any $\boldsymbol{X}^\star$ make the associated constraint saturated is also not restrictive. For example, we prove in *SI Appendix* that this condition is satisfied for the cross-entropy loss. We also remark that Eq. **15** is not the unique convex relaxation. An alternative is to relax Eq. **7** via a nuclear norm-constrained convex program (56), (57) (see more details in *SI Appendix*).

Minority Collapse. With the technique of convex relaxation in place, now we numerically solve the Layer-Peeled Model on imbalanced datasets, with the goal of identifying possible nontrivial patterns. As a worthwhile starting point, we consider a dataset that has K_A majority classes, each containing n_A training examples, and K_B minority classes, each containing n_B training examples. That is, assume $n_1 = n_2 = \cdots = n_{K_A} = n_A$ and $n_{K_A+1} = n_{K_A+2} = \cdots = n_K = n_B$. For convenience, call $R := n_A/n_B > 1$ the imbalance ratio. Note that the case $R = 1$ reduces to the balanced setting.

An important question is to understand how the K_B last-layer minority classifiers behave as the imbalance ratio R increases, as this is directly related to the model performance on the minority classes. To address this question, we show that the average cosine of the angles between any pair of the K_B minority classifiers in Fig. 3 by solving the simple convex program **[15]**. This figure reveals a two-phase behavior of the minority classifiers $\boldsymbol{w}_{K_A+1}^\star, \boldsymbol{w}_{K_A+2}^\star, \ldots, \boldsymbol{w}_K^\star$ as R increases:

1. When $R < R_0$ for some $R_0 > 0$, the average between-minority-class angle becomes smaller as R increases.
2. Once $R \geq R_0$, the average between-minority-class angle becomes zero, and, in addition, the minority classifiers have about the same length. This implies that all the minority classifiers collapse to a single vector.

Above, the phase transition point R_0 depends on the class sizes K_A, K_B and the thresholds E_H, E_W. This value becomes smaller when E_W, E_H, or the number of minority classes K_B is smaller while fixing the other parameters (see more numerical examples in *SI Appendix*, Fig. S2).

We refer to the phenomenon that appears in the second phase as Minority Collapse. While it can be expected that the minority classifiers become closer to each other as the level of imbalance increases, surprisingly, these classifiers become completely indistinguishable once R hits a *finite* value. Once Minority Collapse takes place, the neural network would predict equal probabilities for all the minority classes, regardless of the input. As such, its predictive ability is by no means better than a coin toss when conditioned on the minority classes. This situation would only get worse in the presence of adversarial perturbations. This phenomenon is especially detrimental when the minority classes are more frequent in the application domains than in the training data. Even outside the regime of Minority Collapse, the classification might still be unreliable if the imbalance ratio is large, as the softmax predictions for the minority classes can be close to each other.

To put the observations in Fig. 3 on a firm footing, we prove in the theorem below that Minority Collapse indeed emerges in the Layer-Peeled Model as R tends to infinity.

Theorem 5. *Assume $p \geq K$ and $n_A/n_B \to \infty$, and fix K_A and K_B. Let $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ be any global minimizer of the Layer-Peeled Model (Eq.* **7**) *with the cross-entropy loss. As $R \equiv n_A/n_B \to \infty$, we have*

$$\lim \boldsymbol{w}_k^\star - \boldsymbol{w}_{k'}^\star = \boldsymbol{0}_p,\ \text{for all} K_A < k < k' \leq K.$$

[#] Although Eq. **15** involves a semidefinite constraint, it is not a semidefinite program in the strict sense because a semidefinite program uses a linear objective function.

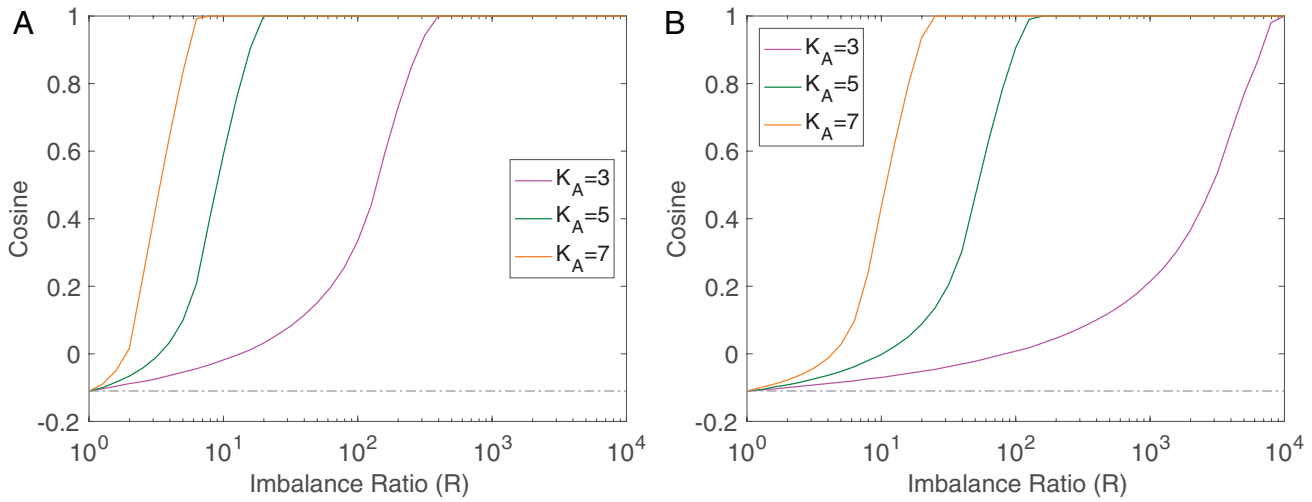

Fig. 3. The average cosine of the angles between any pair of the minority classifier solved from the Layer-Peeled Model. The average cosine reaches 1 once R is above some threshold. The total number of classes $K_A + K_B$ is fixed to 10. The gray dash-dotted line indicates the value of $-\frac{1}{K-1}$, which is given by Eq. **10**. The between-majority-class angles can still be large, even when Minority Collapse emerges. Notably, our simulation suggests that the minority classifiers exhibit an equiangular frame, and so do the majority classifiers. (*A*) $E_W = 1$, $E_H = 5$. (*B*) $E_W = 1$, $E_H = 10$.

To intuitively see why Minority Collapse occurs, first note that the majority classes become the predominant part of the risk function as the level of imbalance increases. The minimization of the objective, therefore, pays too much emphasis on the majority classifiers, encouraging the between-majority-class angles to grow and meanwhile shrinking the between-minority-class angles to zero. As an aside, an interesting question for future work is to prove that $\boldsymbol{w}_k^\star$ and $\boldsymbol{w}_{k'}^\star$ are exactly equal for sufficiently large R.

Experiments. At the moment, Minority Collapse is merely a prediction of the Layer-Peeled Model. An immediate question thus is: Does this phenomenon really occur in real-world neural networks? At first glance, it does not necessarily have to be the case since the Layer-Peeled Model is a dramatic simplification of deep neural networks.

To address this question, we resort to computational experiments.[##] Explicitly, we consider training two network architectures, VGG and ResNet (58), on the FashionMNIST (59) and CIFAR10 datasets and, in particular, replace the dropout layers in VGG with batch normalization (60). As both datasets have 10 classes, we use three combinations of $(K_A, K_B) = (3, 7), (5, 5), (7, 3)$ to split the data into majority classes and minority classes. In the case of FashionMNIST (CIFAR10), we let the K_A majority classes each contain all the $n_A = 6,000$ ($n_A = 5,000$) training examples from the corresponding class of FashionMNIST (CIFAR10), and the K_B minority classes each have $n_B = 6,000/R$ ($n_B = 5,000/R$) examples randomly sampled from the corresponding class. The rest of the experiment setup is basically the same as ref. (4). In detail, we use the cross-entropy loss and stochastic gradient descent with momentum 0.9 and weight decay $\lambda = 5 \times 10^{-4}$. The networks are trained for 350 epochs with a batch size of 128. The initial learning is annealed by a factor of 10 at $1/3$ and $2/3$ of the 350 epochs. The only difference from ref. (4) is that we simply set the learning rate to 0.1 instead of sweeping over 25 learning rates between 0.0001 and 0.25. This is because the test performance of our trained models is already comparable with their best reported test accuracy. Detailed training and test performance are displayed in *SI Appendix*, Tables S1 and S2.

The results of the experiments above are displayed in Fig. 4. This figure clearly indicates that the angles between the minority classifiers collapse to zero as soon as R is large enough. Moreover, the numerical examination in Table 1 shows that the norm of the classifier is constant across the minority classes. Taken together, these two pieces clearly give evidence for the emergence of Minority Collapse in these neural networks, thereby further demonstrating the effectiveness of our Layer-Peeled Model. Besides, Fig. 4 also shows that the issue of Minority Collapse is compounded when there are more majority classes, which is consistent with Fig. 3.

Next, in order to get a handle on how Minority Collapse impacts the test accuracy, we plot the results of another numerical study in Fig. 5. The setting is the same as Fig. 4, except that now we randomly sample six or five examples per class for the minority classes, depending on whether the dataset is FashionMNIST or CIFAR10. The results show that the performance of the trained model deteriorates in the test data when the imbalance ratio $R = 1,000$, when Minority Collapse has occurred or is about to occur. This is by no means intuitive a priori, as the test performance is only restricted to the minority classes and a large value of R only leads to more training data in the majority classes without affecting the minority classes at all.

It is worthwhile to mention that the emergence of Minority Collapse would prevent the model from achieving zero training error. This is because its prediction is uniform over the minority classes, and, therefore, the "argmax" rule does not give the correct label for a training example from a minority class. As such, the occurrence of Minority Collapse is a departure from the terminal phase of deep-learning training. While this fact seems to contradict conventional wisdom on the approximation power of deep learning, it is important to note that the constraints in the Layer-Peeled Model or, equivalently, weight decay in neural networks limits the expressive power of deep-learning models. Besides, it is equally important to recognize that the training error, which mostly occurs in the minority classes, is actually very small when Minority Collapse emerges since the minority examples only account for a small portion of the entire training set. In this spirit, the aforementioned departure is not as significant as it appears at first glance since the training error is generally, if not always, not exactly zero (see, e.g., ref. (4)). From an optimization point of view, a careful examination indicates that Minority Collapse can be attributed to the two constraints

[##] Our code is publicly available at https://github.com/HornHehhf/LPM.

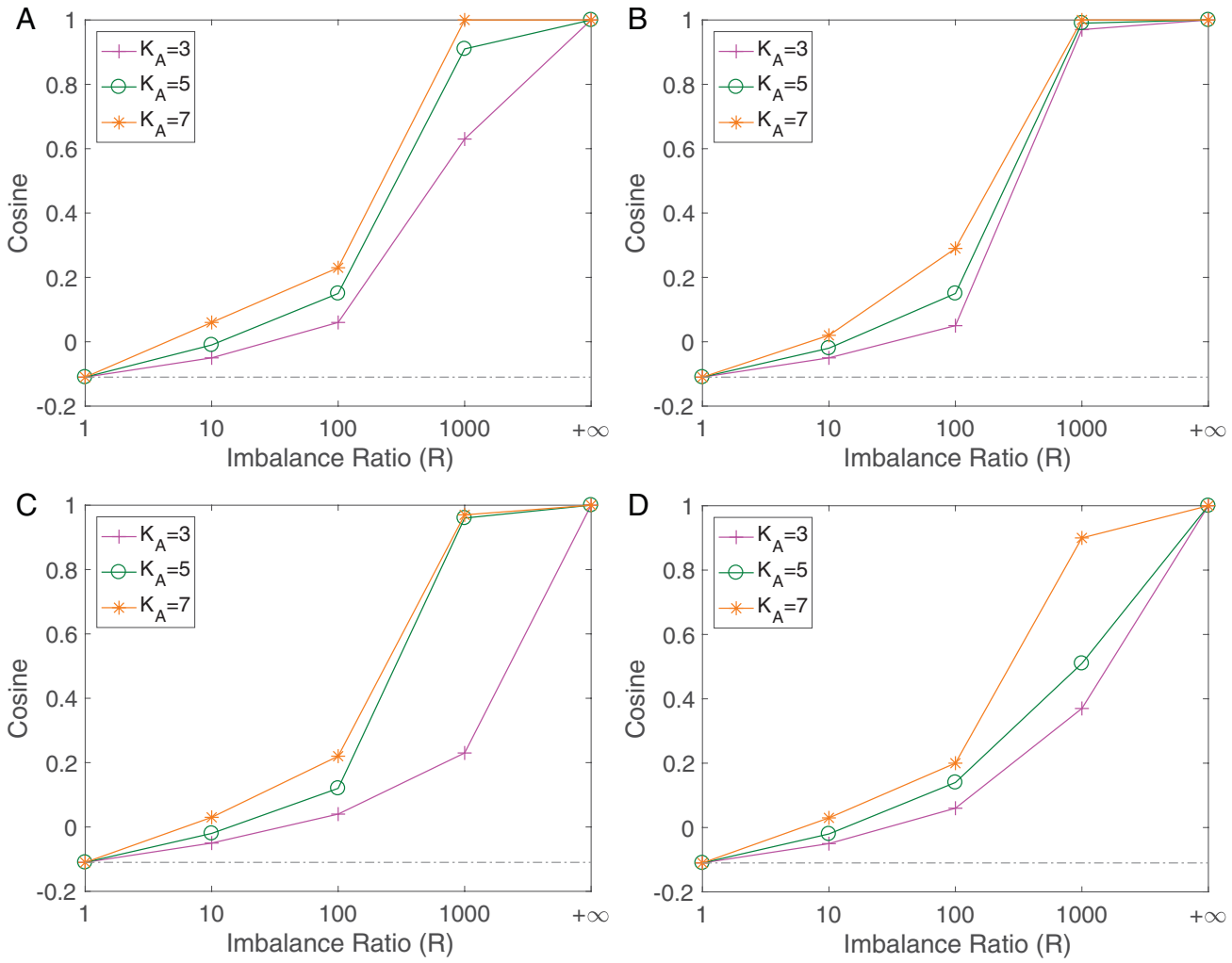

Fig. 4. Occurrence of Minority Collapse in deep neural networks. Each curve denotes the average between-minority-class cosine. We fix $K_A + K_B = 10$. In particular, *B* shares the same setting with Fig. 2in Section 1, where the LPM-based predictions are given by (E_W, E_H) such that the two constraints in the Layer-Peeled Model become active for the weights of the trained networks. For ResNet 18, Minority Collapse also occurs as long as *R* is sufficiently large. Specifically, the average cosine would hit 1 for $K_A = 7$ when $R = 5,000$ on CIFAR10, and when $R = 3,000$ on FashionMNIST. (*A*) VGG11 on FashionMNIST. (*B*) VGG13 on CIFAR10. (*C*) ResNet18 on FashionMNIST. (*D*) ResNet18 on CIFAR10.

in the Layer-Peeled Model or the ℓ_2 regularization in Eq. **1**. For example, Fig. 2 shows that Minority Collapse occurs earlier with a larger value of λ. However, this issue does not disappear by simply setting a small penalty coefficient λ, as the imbalance ratio can be arbitrarily large.

How to Mitigate Minority Collapse?

In this section, we further exploit the use of the Layer-Peeled Model in an attempt to lessen the detrimental effect of Minority Collapse. Instead of aiming to develop a full set of methodologies to overcome this issue, which is beyond the scope of the paper, our aim is to evaluate some simple techniques used for imbalanced datasets.

Among many approaches to handling class imbalance in deep learning (see the review in ref. (11)), perhaps the most popular one is to oversample training examples from the minority classes (61–64). In its simplest form, this sampling scheme retains all majority training examples while duplicating each

Table 1. Variability of the lengths of the minority classifiers when $R = \infty$

Dataset	FashionMNIST						CIFAR10					
Network architecture	VGG11			ResNet18			VGG13			ResNet18		
No. of majority classes	$K_A = 3$	$K_A = 5$	$K_A = 7$	$K_A = 3$	$K_A = 5$	$K_A = 7$	$K_A = 3$	$K_A = 5$	$K_A = 7$	$K_A = 3$	$K_A = 5$	$K_A = 7$
Norm variation	2.7×10^{-5}	4.4×10^{-8}	6.0×10^{-8}	1.4×10^{-5}	5.0^{-8}	6.3×10^{-8}	1.4×10^{-4}	9.0×10^{-7}	5.2×10^{-8}	5.4×10^{-5}	3.5×10^{-7}	5.4×10^{-8}

Each number in the row of "norm variation" is $\mathrm{Std}(\|w_B^\star\|)/\mathrm{Avg}(\|w_B^\star\|)$, where $\mathrm{Std}(\|w_B^\star\|)$ denotes the SD of the lengths of the K_B classifiers and the denominator denotes the average. The results indicate that the classifiers of the minority classes have almost the same length.

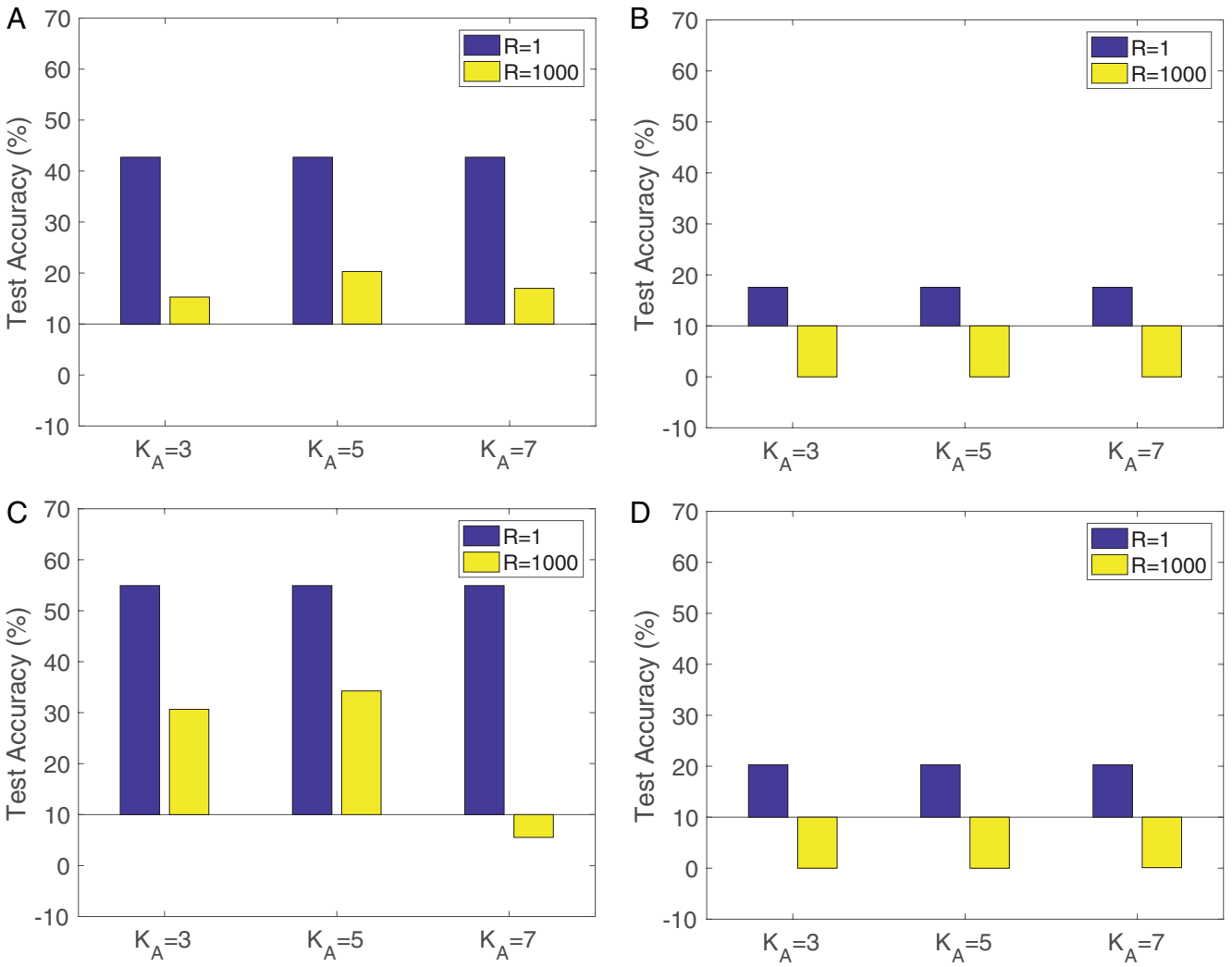

Fig. 5. Comparison of the test accuracy on the minority classes between $R = 1$ and $R = 1,000$. We fix $K_A + K_B = 10$ and use $n_B = 6$ ($n_B = 5$) training examples from each minority class and $n_A = 6R$ ($n_A = 5R$) training examples from each majority class in FashionMNIST (CIFAR10). Note that when $R = 1,000$, the test accuracy on the minority classes can be lower than 10% because the trained neural networks misclassify many examples in the minority classes as some majority classes. (*A*) VGG11 on FashionMNIST. (*B*) VGG13 on CIFAR10. (*C*) ResNet18 on FashionMNIST. (*D*) ResNet18 on CIFAR10.

training example from the minority classes for w_r times, where the oversampling rate w_r is a positive integer. Oversampling in effect transforms the original problem to the minimization of an optimization problem by replacing the risk term in Eq. **1** with

$$\frac{1}{n_A K_A + w_r n_B K_B}\left[\sum_{k=1}^{K_A}\sum_{i=1}^{n_A}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}_{k,i};\boldsymbol{W}_{\text{full}}),\boldsymbol{y}_k) + w_r\sum_{k=K_A+1}^{K}\sum_{i=1}^{n_B}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}_{k,i};\boldsymbol{W}_{\text{full}}),\boldsymbol{y}_k)\right], \quad \textbf{[17]}$$

while keeping the penalty term $\frac{\lambda}{2}\|\boldsymbol{W}_{\text{full}}\|^2$. Note that oversampling is closely related to weight adjusting (see more discussion in *SI Appendix*).

A close look at Eq. **17** suggests that the neural network obtained by minimizing this program might behave as if it were trained on a (larger) dataset with n_A and $w_r n_B$ examples in each majority class and minority class, respectively. To formalize this intuition, as earlier, we start by considering the Layer-Peeled Model in the case of oversampling:

$$\min_{\boldsymbol{H},\boldsymbol{W}}\ \frac{1}{N'}\left[\sum_{k=1}^{K_A}\sum_{i=1}^{n_A}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k) + w_r\sum_{k=K_A+1}^{K}\sum_{i=1}^{n_B}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k)\right]$$
$$\text{s.t.}\ \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W,$$
$$\frac{1}{K}\sum_{k=1}^{K_A}\frac{1}{n_A}\sum_{i=1}^{n_A}\|\boldsymbol{h}_{k,i}\|^2 + \frac{1}{K}\sum_{k=K_A+1}^{K}\frac{1}{n_B}\sum_{i=1}^{n_B}\|\boldsymbol{h}_{k,i}\|^2 \le E_H, \quad \textbf{[18]}$$

where $N' := n_A K_A + w_r n_B K_B$.

The following result confirms our intuition that oversampling indeed boosts the size of the minority classes for the Layer-Peeled Model.

Proposition 1. *Assume $p \ge 2K$ and the loss function $\mathcal{L}$ is convex in the first argument. Let $\boldsymbol{X}^\star$ be any minimizer of the convex program* **[15]** *with $n_1 = n_2 = \cdots = n_{K_A} = n_A$ and $n_{K_A+1} = n_{K_A+2} = \cdots = n_K = w_r n_B$. Define $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ as*

$$\left[\boldsymbol{h}_1^\star, \boldsymbol{h}_2^\star, \ldots, \boldsymbol{h}_K^\star, (\boldsymbol{W}^\star)^\top\right] = \boldsymbol{P}(\boldsymbol{X}^\star)^{1/2},$$
$$\boldsymbol{h}_{k,i}^\star = \boldsymbol{h}_k^\star, \text{ for all} 1 \le i \le n_A, 1 \le k \le K_A,$$
$$\boldsymbol{h}_{k,i}^\star = \boldsymbol{h}_k^\star, \text{ for all} 1 \le i \le n_B, K_A < k \le K, \quad \textbf{[19]}$$

where $\boldsymbol{P} \in \mathbb{R}^{p \times 2K}$ *is any partial orthogonal matrix such that* $\boldsymbol{P}^\top \boldsymbol{P} = \boldsymbol{I}_{2K}$. *Then,* $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ *is a global minimizer of the oversampling-adjusted Layer-Peeled Model* (*Eq.* **18**). *Moreover, if all* $\boldsymbol{X}^\star$*'s satisfy* $\frac{1}{K}\sum_{k=1}^K \boldsymbol{X}^\star(k,k) = E_H$, *then all the solutions of Eq.* **18** *are in the form of Eq.* **19**.

Together with Lemma 1, Proposition 1 shows that the number of training examples in each minority class is now in effect $w_r n_B$ instead of n_B in the Layer-Peeled Model. In the special case $w_r = n_A/n_B \equiv R$, the results show that all the angles are equal between any given pair of the last-layer classifiers, no matter if they fall in the majority or minority classes.

We turn to Fig. 6 for an illustration of the effects of oversampling on real-world deep-learning models, using the same experimental setup as in Fig. 5. From Fig. 6, we see that the angles between pairs of the minority classifiers become larger as the oversampling rate w_r increases. Consequently, the issue of Minority Collapse becomes less detrimental in terms of training accuracy as w_r increases. This again corroborates the predictive ability of the Layer-Peeled Model.

Next, we refer to Table 2 for effect on the test performance. The results clearly demonstrate the improvement in test accuracy using oversampling, with certain choices of the oversampling rate. The improvement is noticeable on both the minority classes and all classes.

Behind the results of Table 2, however, it reveals an issue when addressing Minority Collapse by oversampling. Specifically, this technique might lead to degradation of test performance using a very large oversampling rate w_r, which, though, can mitigate Minority Collapse. How can we efficiently select an oversampling rate for optimal test performance? More broadly, Minority Collapse does not seem likely to be fully resolved by sampling-based approaches alone, and the doors are wide open for future investigation.

Discussion

In this paper, we have developed the Layer-Peeled Model as a simple, yet effective, modeling strategy toward understanding well-trained deep neural networks. The derivation of this model follows a top-down strategy by isolating the last layer from the remaining layers. Owing to the analytical and numerical tractability of the Layer-Peeled Model, we provide some explanation of a recently observed phenomenon called neural collapse in deep neural networks trained on balanced datasets (4). Moving to imbalanced datasets, an analysis of this model suggests that

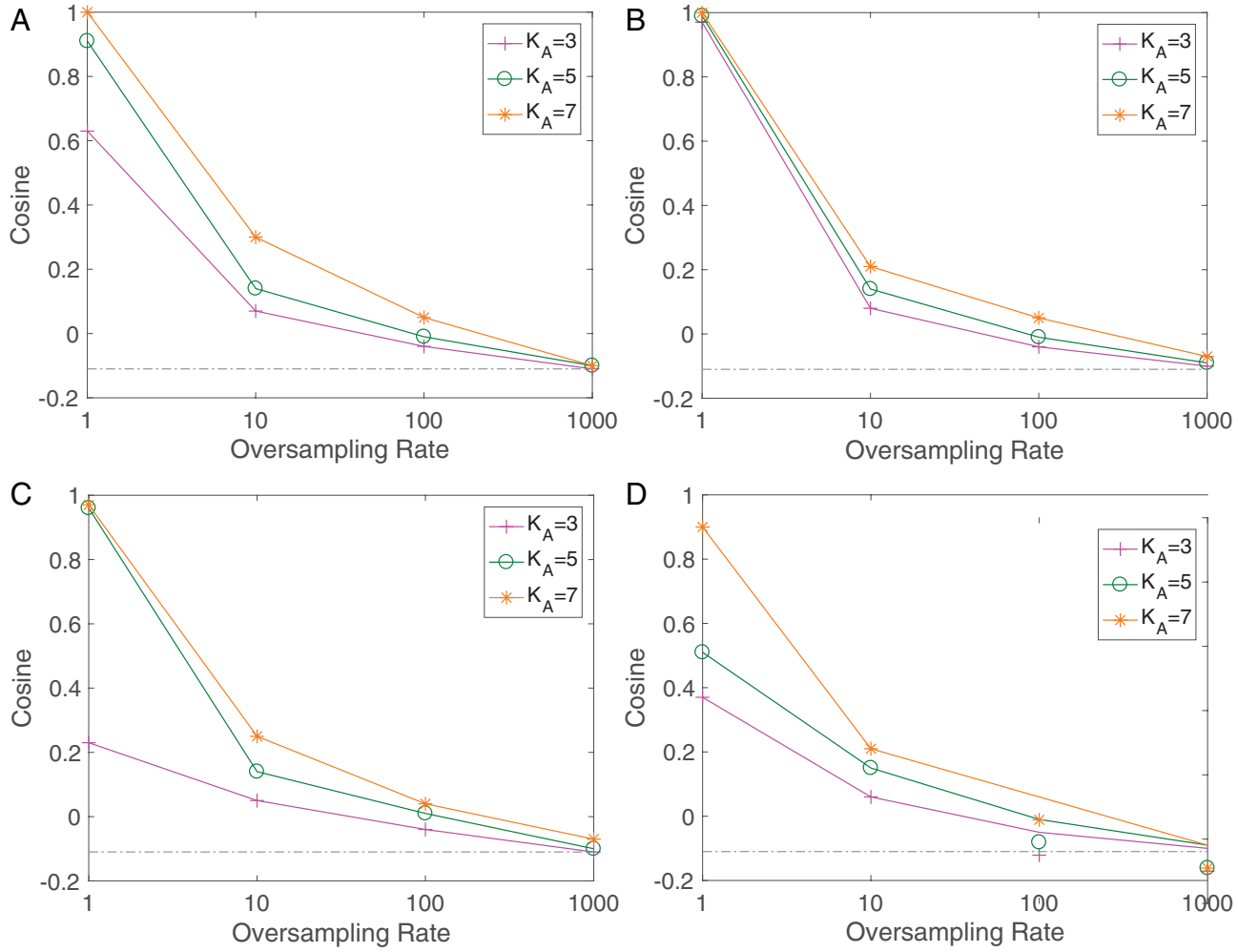

Fig. 6. Effect of oversampling when the imbalance ratio is $R = 1,000$. Each plot shows the average cosine of the between-minority-class angles. The results indicate that increasing the oversampling rate would enlarge the between-minority-class angles. (*A*) VGG11 on FashionMNIST. (*B*) VGG13 on CIFAR10. (*C*) ResNet18 on FashionMNIST. (*D*) ResNet18 on CIFAR10.

Table 2. Test accuracy (%) on FashionMNIST when $R = 1,000$

Network architecture	VGG11			ResNet18		
No. of majority classes	$K_A = 3$	$K_A = 5$	$K_A = 7$	$K_A = 3$	$K_A = 5$	$K_A = 7$
Original (minority)	15.29	20.30	17.00	30.66	34.26	5.53
Oversampling (minority)	41.13	57.22	30.50	37.86	53.46	8.13
Improvement (minority)	25.84	36.92	13.50	7.20	19.20	2.60
Original (overall)	40.10	57.61	69.09	50.88	64.89	66.13
Oversampling (overall)	58.25	76.17	73.37	55.91	74.56	67.10
Improvement (overall)	18.15	18.56	4.28	5.03	9.67	0.97

For example, "Original (minority)" means that the test accuracy is evaluated only on the minority classes, and oversampling is not used. When oversampling is used, we report the best test accuracy among four oversampling rates: 1, 10, 100, and 1,000. The best test accuracy is never achieved at $w_r = 1,000$, indicating that oversampling with a large w_r would impair the test performance.

the last-layer classifiers corresponding to the minority classes would collapse to a single vector once the imbalance level is above a certain threshold. This phenomenon, whic h we refer to as Minority Collapse, occurs consistently in our computational experiments.

The efficacy of the Layer-Peeled Model in analyzing well-trained deep-learning models implies that the ansatz Eq. **6**—a crucial step in the derivation of this model—is at least a useful approximation. Moreover, this ansatz can be further justified by the following result in an indirect manner, which, together with Theorem 1, shows that the ℓ_2 norm suggested by the ansatz happens to be the only choice among all the ℓ_q norms that is consistent with empirical observations. Its proof is given in *SI Appendix*.

Proposition 2. *Assume $K \geq 3$ and $p \geq K$.*[###] *For any $q \in (0,2) \cup (2,\infty)$, consider the optimization problem*

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k) \\ \text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \leq E_W, \\ & \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{h}_{k,i}\|_q^q \leq E_H, \end{aligned} \qquad [20]$$

where $\mathcal{L}$ is the cross-entropy loss. Then, any global minimizer of this program does not satisfy Eq. **9** *for any positive numbers C and C'. That is, neural collapse does not emerge in this model.*

While the Layer-Peeled Model has demonstrated its noticeable effectiveness, it requires future investigation for consolidation and extension. First, an analysis of the gap between the Layer-Peeled Model and well-trained deep-learning models would be a welcome advance. For example, how does the gap depend on the neural network architectures? How to take into account the sparsity of the last-layer features when using the ReLU activation function? From a different angle, a possible extension is to retain multiple layers following the top-down viewpoint. Explicitly, letting $1 \leq m < L$ be the number of the top layers we wish to retain in the model, we can represent the prediction of the neural network as $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{W}_{\text{full}}) = \boldsymbol{f}(\boldsymbol{h}(\boldsymbol{x}; \boldsymbol{W}_{1:(L-m)}), \boldsymbol{W}_{(L-m+1):L})$ by letting $\boldsymbol{W}_{1:(L-m)}$ and $\boldsymbol{W}_{(L-m+1):L}$ be the first $L-m$ layers and the last m layers, respectively. Consider the m-Layer-Peeled Model:

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{f}(\boldsymbol{h}_{k,i}, \boldsymbol{W}_{(L-m+1):L}), \boldsymbol{y}_k) \\ \text{s.t.} \quad & \frac{1}{K}\|\boldsymbol{W}_{(L-m+1):L}\|^2 \leq E_W, \\ & \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \leq E_H. \end{aligned}$$

The two constraints might be modified to take into account the network architectures. An immediate question is whether this model with $m = 2$ is capable of capturing new patterns of deep-learning training.

From a practical standpoint, the Layer-Peeled Model together with its convex relaxation Eq. **15** offers an analytical and computationally efficient technique to identify and mitigate bias induced by class imbalance. An interesting question is to extend Minority Collapse from the case of two-valued class sizes to general imbalanced datasets. Next, as suggested by our findings in Section 5, how should we choose loss functions in order to mitigate Minority Collapse (64)? Last, a possible use case of the Layer-Peeled Model is to design more efficient sampling schemes to take into account fairness considerations (65–67).

Broadly speaking, insights can be gained not only from the Layer-Peeled Model, but also from its modeling strategy. The details of empirical deep-learning models, though formidable, can often be simplified by rendering a certain part of the network *modular*. When the interest is about the top few layers, for example, this paper clearly demonstrates the benefits of taking a top-down strategy for modeling neural networks, especially in consolidating our understanding of previous results and in discovering new patterns. Owing to its mathematical convenience, the Layer-Peeled Model shall open the door for future research extending these benefits.

Data Availability. All study data are included in the article and/or supporting information. Our code is publicly available at GitHub (https://github.com/HornHehhf/LPM).

ACKNOWLEDGMENTS. We are grateful to X. Y. Han for helpful discussions and feedback on an early version of the manuscript. We thank Gang Wen and Qinqing Zheng for helpful comments. We thank the two anonymous referees for their constructive comments that helped improve the presentation of this work. This work was supported in part by NIH Grant RF1AG063481; NSF Grants CAREER DMS-1847415 and CCF-1934876; an Alfred Sloan Research Fellowship; and the Wharton Dean's Research Fund.

[###] See discussion in the case $K = 2$ in *SI Appendix*.

1. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).
2. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
3. D. Silver *et al.*, Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
4. V. Papyan, X. Y. Han, D. L. Donoho, Prevalence of neural collapse during the terminal phase of deep learning training. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 24652–24663 (2020).
5. A. R. Webb, D. Lowe, The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis. *Neural Netw.* **3**, 367–375 (1990).
6. D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, N. Srebro, The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.* **19**, 2822–2878 (2018).
7. S. Oymak, M. Soltanolkotabi, Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE J. on Sel. Areas Inf. Theory* **1**, 84–105 (2020).

8. Y. Yu, K. H. R. Chan, C. You, C. Song, Y. Ma, Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Adv. Neural Inf. Process. Syst.***33**, 9422–9434 (2020).
9. O. Shamir, Gradient methods never overfit on separable data. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2007.00028 (Accessed 20 December 2020).
10. T. Chen, S. Kornblith, M. Norouzi, G. Hinton, "A simple framework for contrastive learning of visual representations" in *Proceedings of the 37th International Conference on Machine Learning*, H. Daumé III, A. Singh, Eds. (Proceedings of Machine Learning Research, Vienna, Austria, 2020), **vol. 119**, pp. 1597–1607.
11. J. M. Johnson, T. M. Khoshgoftaar, Survey on deep learning with class imbalance. *J. Big Data* **6**, 27 (2019).
12. A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, University of Toronto, Toronto, Canada (2009).
13. K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition" in *ICLR 2015: 3rd International Conference on Learning Representations*, Y. Bengio, Y. LeCun, Eds. (ICLR, San Diego, CA, 2015).
14. D. Yarotsky, Error bounds for approximations with deep ReLU networks. *Neural Netw.* **94**, 103–114 (2017).
15. A. Jacot, F. Gabriel, C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks" in *NeurIPS 2018: 32nd Conference on Neural Information Processing Systems*, S. Bengio *et al*., Eds. (Curran Associates, Inc., Montreal, Canada, 2018), **vol. 31**, pp. 8580–8589.
16. S. S. Du, J. D. Lee, H. Li, L. Wang, X. Zhai, "Gradient descent finds global minima of deep neural networks" in *Proceedings of the 36th International Conference on Machine Learning* , K. Chaudhuri, R. Salakhutdinov, Eds. (Proceedings of Machine Learning Research, Long Beach, CA, 2019), pp. 1675–1685.
17. Z. Allen-Zhu, Y. Li, Z. Song, "A convergence theory for deep learning via over-parameterization" in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri, R. Salakhutdinov, Eds. (Proceedings of Machine Learning Research, Long Beach, CA, 2019), pp. 2388–2464.
18. D. Zou, Y. Cao, D. Zhou, Q. Gu, Stochastic gradient descent optimizes overparameterized deep ReLU networks. *Mach. Learn.*, **109**, 467–492 (2020).
19. L. Chizat, E. Oyallon, F. Bach, "On lazy training in differentiable programming" in *Advances in Neural Information Processing Systems*, H. Wallach et al., Eds. (Curran Associates, Inc., Vancouver, Canada, 2019), pp. 2937–2947.
20. E. Weinan, C. Ma, L. Wu, A comparative analysis of the optimization and generalization property of two-layer neural network and random feature models under gradient descent dynamics. *arXiv* [Preprint] (2020). https://arxiv.org/abs/1904.04326 (Accessed 21 December 2020).
21. P. Bartlett, D. Foster, M. Telgarsky, Spectrally-normalized margin bounds for neural networks. *Adv. Neural Inf. Process. Syst.* **30**, 6241–6250 (2017).
22. H. He, W. J. Su, "The local elasticity of neural networks" in *International Conference on Learning Representations* (OpenReview.net, Addis Ababa, Ethiopia, 2020).
23. T. Poggio, A. Banburski, Q. Liao, Theoretical issues in deep networks. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 30039–30045 (2020).
24. S. Mei, A. Montanari, P. M. Nguyen, A mean field view of the landscape of two-layer neural networks. *Proc. Natl. Acad. Sci. U.S.A.* **115**, E7665–E7671 (2018).
25. J. Sirignano, K. Spiliopoulos, Mean field analysis of neural networks: A central limit theorem. *Stoch. Process. Appl.* **130**, 1820–1852 (2020).
26. G. M. Rotskoff, E. Vanden-Eijnden, "Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks," in *Advances in Neural Information Processing Systems*, S. Bengio *et al*., Eds. (Curran Associates, Inc., Montreal, Canada, 2018), pp. 7146–7155.
27. C. Fang, J. D. Lee, P. Yang, T. Zhang, Modeling from features: A mean-field framework for over-parameterized deep neural networks. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2007.01452 (Accessed 23 December 2020).
28. R. Kuditipudi *et al.*, "Explaining landscape connectivity of low-cost solutions for multilayer nets" in *Advances in Neural Information Processing Systems* H. Wallach *et al*., Eds. (Curran Associates, Inc., Vancouver, Canada, 2019), pp. 14601–14610.
29. B. Shi, W. J. Su, M. I. Jordan, On learning rates and Schrödinger operators. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2004.06977 (Accessed 26 December 2020).
30. C. Fang, H. Dong, T. Zhang, Mathematical models of overparameterized neural networks. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2012.13982 (Accessed 5 January 2021).
31. F. He, D. Tao, Recent advances in deep learning theory. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2012.10931 (Accessed 5 January 2021).
32. J. Fan, C. Ma, Y. Zhong, A selective overview of deep learning. *arXiv* [Preprint] (2019). https://arxiv.org/abs/1904.05526 (Accessed 26 December 2020).
33. R. Sun, Optimization for deep learning: Theory and algorithms. *arXiv* [Preprint] (2019). https://arxiv.org/abs/1912.08957 (Accessed 26 December 2020).
34. T. Strohmer, R. W. Heath, Grassmannian frames with applications to coding and communication. *Appl. Comput. Harmon. Anal.* **14**, 257–275 (2003).
35. S. Ma, R. Bassily, M. Belkin, "The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning" in *Proceedings of the 35th International Conference on Machine Learning* J. Dy, A. Krause, Eds. (Proceedings of Machine Learning Research, Stockholm, Sweden, 2018), vol. 80, pp. 3325–3334.
36. M. Belkin, D. Hsu, S. Ma, S. Mandal, Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 15849–15854 (2019).
37. T. Liang, A. Rakhlin, Just interpolate: Kernel "ridgeless" regression can generalize. *Ann. Stat.* **48**, 1329–1347 (2020).
38. P. L. Bartlett, P. M. Long, G. Lugosi, A. Tsigler, Benign overfitting in linear regression. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 30063–30070 (2020).
39. Z. Li, W. Su, D. Sejdinovic, Benign overfitting and noisy features. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2008.02901 (Accessed 27 December 2020).
40. D. G. Mixon, H. Parshall, J. Pi, Neural collapse with unconstrained features. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2011.11619 (Accessed 27 December 2020).
41. E. Weinan, S. Wojtowytsch, On the emergence of tetrahedral symmetry in the final and penultimate layers of neural network classifiers. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2012.05420 (Accessed 6 January 2021).
42. J. Lu, S. Steinerberger, Neural collapse with cross-entropy loss. *arXiv* [Preprint] (2021). https://arxiv.org/abs/2012.08465 (Accessed 6 January 2021).
43. T. Ergen, M. Pilanci, Convex duality of deep neural networks. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2002.09773 (Accessed 27 December 2020).
44. T. Poggio, Q. Liao, Explicit regularization and implicit bias in deep network classifiers trained with the square loss. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2101.00072 (Accessed 10 January 2021).
45. J. Pennington, R. Socher, C. D. Manning, "GloVe: Global vectors for word representation" in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Doha, Qatar, 2014), pp. 1532–1543.
46. N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, H. Khandeparkar, "A theoretical analysis of contrastive unsupervised representation learning" in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri, R. Salakhutdinov, Eds. (Proceedings of Machine Learning Research, Long Beach, CA, 2019), **vol. 97**, pp. 5628–5637.
47. A. Baevski, H. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2006.11477 (Accessed 29 December 2020).
48. P. Khosla *et al.*, Supervised contrastive learning. *arXiv* [Preprint] (2020). https://arxiv.org/abs/2004.11362 (Accessed 29 December 2020).
49. A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library" in Advances in *Neural Information Processing Systems* H. Wallach *et al*., Eds. (Curran Associates, Inc., Vancouver, Canada, 2019), pp. 8026–8037.
50. S. Wang *et al.*, "Training deep neural networks on imbalanced data sets" in 2016 International Joint Conference on Neural Networks (IJCNN) (IEEE, Vancouver, Canada, 2016), pp. 4368–4374.
51. C. Huang, Y. Li, C. C. Loy, X. Tang, Learning deep representation for imbalanced classification in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, Las Vegas, NV, 2016), pp. 5375–5384.
52. K. Madasamy, M. Ramaswami, Data imbalance and classifiers: Impact and solutions from a big data perspective. *Int. J. Comput. Intell. Res* **13**, 2267–2281 (2017).
53. B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, A. Oliva, Places: An image database for deep scene understanding. *arXiv* [Preprint] (2016). https://arxiv.org/abs/1610.02055 (Accessed 2 January 2021).
54. E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, R. Weischedel, "Ontonotes: The 90% solution" in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers* (Association for Computational Linguistics, New York, NY, 2006), pp. 57–60.
55. J. F. Sturm, S. Zhang, On cones of nonnegative quadratic functions. *Math. Oper. Res.* **28**, 246–267 (2003).
56. F. Bach, J. Mairal, J. Ponce, Convex sparse matrix factorizations. *arXiv* [Preprint] (2008). https://arxiv.org/abs/0812.1869 (Accessed 3 January 2021).
57. B. D. Haeffele, R. Vidal, Structured low-rank matrix factorization: Global optimality, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**, 1468–1482 (2020).
58. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, Las Vegas, NV, 2016), pp. 770–778.
59. H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv* [Preprint] (2017). https://arxiv.org/abs/1708.07747 (Accessed 3 January 2021).
60. S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift" in *International Conference on Machine Learning*, F. Bach, D. Blei, Eds. (PMLR, Lille, France, 2015), pp. 448–456.
61. M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **106**, 249–259 (2018).
62. J. Shu *et al.*, "Meta-weight-net: Learning an explicit mapping for sample weighting" in *NeurIPS 2019: 33rd International Conference on Neural Information Processing Systems*, H. Wallach *et al.*, Eds. (Advances in Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, 2019), **vol. 32**, pp. 1919–1930.
63. Y. Cui, M. Jia, T. Y. Lin, Y. Song, S. Belongie, "Class-balanced loss based on effective number of samples" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Long Beach, CA, 2019), pp. 9268–9277.
64. K. Cao, C. Wei, A. Gaidon, N. Arechiga, T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss" in *NeurIPS 2019: 33rd International Conference on Neural Information Processing Systems*, H. Wallach *et al*., Eds. (Advances in Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, pp. 1567–1578.
65. J. Buolamwini, T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification" in *Proceedings of the First Conference on Fairness, Accountability and Transparency* S. A. Friedler, C. Wilson, Eds. (Proceedings of Machine Learning Research, New York, NY, 2018), vol. 81, pp. 77–91.
66. J Zou, L Schiebinger, AI can be sexist and racist—It's time to make it fair. *Naturel* **559**, 324–326 (2018).
67. N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning. *arXiv* [Preprint] (2019). https://arxiv.org/abs/1908.09635 (Accessed 4 January 2021).