# SYMMETRIC NEURAL-COLLAPSE REPRESENTATIONS WITH SUPERVISED CONTRASTIVE LOSS: THE IMPACT OF RELU AND BATCHING

**Ganesh Ramachandra Kini**[‡], **Vala Vakilian**[†], **Tina Behnia**[†], **Jaidev Gill**[†], **Christos Thrampoulidis**[†]

[‡]University of California, Santa Barbara, USA

[†]University of British Columbia, Canada *

## ABSTRACT

Supervised contrastive loss (SCL) is a competitive and often superior alternative to the cross-entropy loss for classification. While prior studies have demonstrated that both losses yield symmetric training representations under balanced data, this symmetry breaks under class imbalances. This paper presents an intriguing discovery: the introduction of a ReLU activation at the final layer effectively restores the symmetry in SCL-learned representations. We arrive at this finding analytically, by establishing that the global minimizers of an unconstrained features model with SCL loss and entry-wise non-negativity constraints form an orthogonal frame. Extensive experiments conducted across various datasets, architectures, and imbalance scenarios corroborate our finding. Importantly, our experiments reveal that the inclusion of the ReLU activation restores symmetry without compromising test accuracy. This constitutes the first geometry characterization of SCL under imbalances. Additionally, our analysis and experiments underscore the pivotal role of batch selection strategies in representation geometry. By proving necessary and sufficient conditions for mini-batch choices that ensure invariant symmetric representations, we introduce batch-binding as an efficient strategy that guarantees these conditions hold.

## 1 INTRODUCTION

The prevalence of deep-neural networks (DNNs) has led to a growing research interest in understanding their underlying mechanisms. A recent research thread, focusing on classification tasks, explores whether it is possible to describe the structure of weights learned by DNNs when trained beyond zero-training error. The specific characteristics of this structure will depend on the DNN being used, the dataset being trained on, and the chosen optimization hyperparameters. Yet, *is it possible to identify macroscopic structural characteristics that are common among these possibilities?*

In an inspiring study, Papyan et al. (2020) demonstrates this is possible for the classifiers and for the embeddings when training with cross-entropy (CE) loss and balanced datasets. Through extensive experiments over multiple architectures and datasets with an equal number of examples per class, they found that the geometries of classifiers and of centered class-mean embeddings (outputs of the last hidden layer) consistently converge during training to a common simplex equiangular tight frame (ETF), a structure composed of vectors that have equal norms and equal angles between them, with the angles being the maximum possible. Moreover, they observed neural-collapse (NC), a property where embeddings of individual examples from each class converge to their class-mean embedding.

Numerous follow-up studies have delved deeper into explaining this phenomenon and further investigating how the converging geometry changes with class imbalances. The unconstrained-features model (UFM), proposed independently by Mixon et al. (2020); Fang et al. (2021); Graf et al. (2021); Lu & Steinerberger (2020), plays a central role in the majority of these follow-up studies. Specifically,

*This work is supported by an NSERC Discovery Grant, NSF Grant CCF-2009030, and by a CRG8-KAUST award. JG and CT gratefully acknowledge the support of NSERC Undergraduate Student Research Grant. The authors also acknowledge use of the Sockeye cluster by UBC Advanced Research Computing.

the UFM serves as a theoretical abstraction for DNN training, in which the network architecture is viewed as a powerful black-box that generates embeddings without any restrictions in the last (hidden) layer. For CE loss, the UFM minimizes $\min_{\mathbf{w}_c \in \mathbb{R}^d; \mathbf{h}_i \in \mathbb{R}^d} \mathcal{L}_{\text{CE}}\big(\{\mathbf{w}_c\}_{c\in[k]}, \{\mathbf{h}_i\}_{i\in[n]}\big)$ in which both classifiers $\mathbf{w}_c$ for the $k$ classes and embeddings $\mathbf{h}_i$ for the $n$ training examples are unconstrainedly optimized over $\mathbb{R}^d$. Zhu et al. (2021); Graf et al. (2021); Fang et al. (2021) have verified that the global minimum of this non-convex problem satisfies NC and follows the ETF geometry observed in DNN experiments by Papyan et al. (2020). Recent works by Thrampoulidis et al. (2022); Fang et al. (2021); Behnia et al. (2023) have demonstrated that the global optimum of the UFM changes when classes are imbalanced. Yet, the new solution still predicts the geometry observed in DNN experiments, providing evidence that, despite its oversimplification, the UFM is valuable in predicting structural behaviors.

Expanding beyond the scope of CE minimization, Graf et al. (2021) also used the UFM to determine whether optimizing with the supervised-contrastive loss (SCL) results in any alterations to the geometric structure of the learned embeddings.[1] They proved for balanced datasets that the global solution of $\min_{\mathbf{h}_i \in \mathbb{R}^d} \mathcal{L}_{\text{SCL}}\big(\{\mathbf{h}_i\}_{i\in[n]}\big)$ remains a simplex ETF, suggesting that CE and SCL find the same embedding geometries. In Fig. 1, we investigate the geometry of class-means in presence of class-imbalance. The prediction in Graf et al. (2021) only applies to balanced data (SCL with $R = 1$ in Fig. 1) and no prior work has explicitly characterized the geometry of SCL with class imbalances. Observing the middle column of the figure, note that the geometry of embeddings changes drastically[2] as the imbalance ratio $R$ increases. This behavior of SCL is consistent with the CE embeddings geometry, which, as discussed previously, also changes with the imbalance; see last column of Fig. 1.

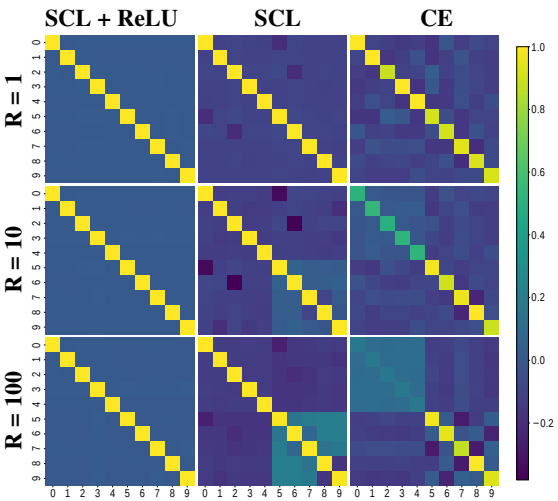



**Figure 1:** Gram matrices, $\mathbf{G}_{\mathbf{M}}/\max_{i;j} |\mathbf{G}_{\mathbf{M}}[i;j]|$, of class-means of learned feature embeddings at last epoch of training, with ResNet-18 on MNIST. **SCL+ReLU:** the mean feature embeddings for different classes are mutually orthogonal, forming an OF, regardless of imbalance (imbalance ratio $R = 1; 10; 100$). This invariance does not hold in the absence of ReLU for **SCL**. Further, **CE** feature geometry is also sensitive to imbalance. The label distribution is STEP imbalanced, with the first five classes as majorities and the rest as minorities. See Sec.E.2 in SM for more information.

In this paper, we arrive at a surprising finding: simply introducing a ReLU activation at the final layer restores the symmetry in SCL-trained representations. This phenomenon is clearly illustrated in the first column of Fig. 1. Below is a detailed description of our contributions.

### 1.1 Summary of contributions

We conduct an in-depth examination of the geometry of training representations (embeddings) of SCL, particularly in relation to varying levels of dataset imbalances. Our study leads us to identify and capture, both analytically and empirically, and for the first time, the impact on the representation geometry of: (i) a straightforward architectural modification, specifically the incorporation of a ReLU activation at the final layer, and (ii) the batch-selection strategy.

**Impact of ReLU.** We find that the addition of a ReLU activation to the last-layer of the architecture results in an orthogonal frame (OF) geometry. That is, a geometry structure composed of class-mean embedding vectors that have equal norms and are mutually orthogonal to each other. A preliminary experimental validation of this finding is illustrated in Fig. 1. The experiments correspond to STEP-imbalanced MNIST data with five majority/minority classes and imbalance ratio $R = 1, 10, 100$. Each heatmap represents the pairwise inner products between the class-means of learned feature embeddings. The figure (first column) reveals that with the addition of ReLU at the last layer, SCL

[1] SCL is designed to train only embeddings and is an extension of the well-known contrastive loss used for unsupervised learning, adapted to supervised datasets (Khosla et al., 2020); see Equation (2).

[2] Analogous finding is reported by Zhu et al. (2022), although not visualized as done here.

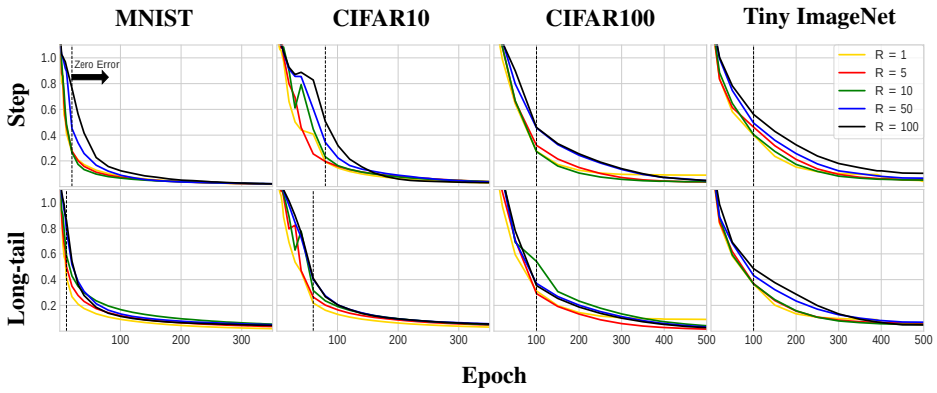



**Figure 2:** Distance of learned embeddings to the OF geometry measured by $\| \frac{\mathbf{G_M}}{\|\mathbf{G_M}\|_F} - \frac{\mathrm{I}_k}{\|\mathrm{I}_k\|_F} \|_F$ across epochs where $\mathbf{G_M} = \mathbf{M}^\top \mathbf{M}$ is the Gram matrix of class-mean embeddings. For MNIST and CIFAR10 we use ResNet-18 and for the more complex CIFAR100 and Tiny-ImageNet, we use ResNet-34 and train using batch-binding (see Sec. 5). Regardless of the imbalance level $R$, the embeddings consistently converge to the OF geometry.

learns orthogonal features *regardless of class imbalance.* This is in stark contrast to the varying geometries of vanilla SCL (optimized commonly without ReLU) and CE loss. Extensive additional experiments for Long-tailed (LT) distributions, other datasets and architectures, are presented in Sec. 3 and also Sec. E in the SM. We also present experimental findings that confirm ReLU symmetrises the geometry without compromising test accuracy.

We support this finding by theoretically investigating the global minimizers of an extended version, denoted as UFM$_+$, of the original unconstrained-features model (UFM). This refinement accounts for the presence of ReLU activations which we impose on the last-layer by minimizing SCL over the non-negative orthant. Concretely, we show that all global solutions of $\min_{\mathbf{h}_i \geq 0} \mathcal{L}_{\mathrm{SCL}}\big(\{\mathbf{h}_i\}_{i\in[n]}\big)$ satisfy NC and the corresponding class-mean features form an OF. This explains the convergence of feature geometry to an OF as consistently observed in our experiments; for example, see Fig. 2.

**Batching and its implications.** Furthermore, we identify the crucial impact of batch selection strategies during SCL training in shaping the learned embedding geometry. Concretely, by analyzing the UFM$_+$, we establish straightforward criteria for the batching scheme, ensuring that any global minimizer of UFM$_+$ forms an OF. These conditions explain the substantial degradation in convergence towards an OF when employing a fixed batch partition instead of randomly reshuffling the batches at each training epoch. Intriguingly, we demonstrate that any arbitrary batching partition can be transformed into one that fulfills our theoretical conditions by performing so-called *batch-binding*. Through extensive experiments in Sec. 5.2 and in the SM, we show that batch-binding consistently accelerates the convergence of the learned geometry towards an OF and even improves convergence to ETF geometry under balanced training in the absence of ReLU.

## 2 Setup

**Notation.** For a positive integer $n$, $[n] := \{1, 2, 3, ..., n\}$. For matrix $\mathbf{V} \in \mathbb{R}^{m\times n}$, $\mathbf{V}[i, j]$ denotes its $(i, j)$-th entry, $\mathbf{v}_j$ denotes the $j$-th column, $\mathbf{V}^\top$ its transpose. We denote $\|\mathbf{V}\|_F$ the Frobenius norm of $\mathbf{V}$. We use $\mathbf{V} \propto \mathbf{X}$ whenever the two matrices are equal up to a scalar constant. We use $\mathbf{V} \geq 0$ to denote the entry-wise non-negativity, i.e., $\mathbf{V}[i, j] \geq 0, \forall i \in [m], j \in [n]$. Finally, $\mathrm{I}_m$ denotes the identity matrix of size $m$.

Consider $k$-class classification and training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in [k]$ represent data samples and corresponding class labels. We use $n_c$ to denote the number of examples in class $c \in [k]$, and $\mathbf{h}(\cdot)$ to denote the last-layer feature-embeddings of a deep-net parameterized by . We compute the SCL on a given batch $B \subseteq [n]$ of training examples as follows,

$$\mathcal{L}_B(\ ) := \sum_{i\in B} \frac{1}{n_{B,y_i} - 1} \sum_{\substack{j\in B \\ y_j = y_i, j\neq i}} \log\left(1 + \sum_{\neq i,j} \exp\left(\frac{1}{\tau}\big(\mathbf{h}(\mathbf{x}_i)^\top \mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{x}_i)^\top \mathbf{h}(\mathbf{x}_j)\big)\right)\right), \quad (1)$$

where $\tau$ is a positive scalar temperature hyper-parameter, and $n_{B;y_i}$ is the number of examples in batch $B$ belonging to class $c = y_i$. To train a deep-net, we minimize SCL (1) over the network parameters on a set of batches $\mathcal{B}$ chosen from the training set. As introduced by Khosla et al. (2020), SCL requires normalized features, so we assume $\|\mathbf{h}(\mathbf{x}_i)\| = 1$. Furthermore, we assume that $d \geq k$ and $n_c, n_{B;c} \geq 2, c \in [k]$.[3] We let $\mathbf{H}_{d\times n} = [\mathbf{h}_1, \cdots, \mathbf{h}_n]$ denote an embeddings matrix where each column corresponds to one of the examples in the training set, i.e., $\mathbf{h}_i := \mathbf{h}(\mathbf{x}_i)$. The *embeddings geometry* or so-called *implicit geometry*[4] refers to the norms and pairwise-angles of these vectors $\mathbf{h}_i, i \in [n]$. Note these quantities correspond exactly to the entries of the Gram matrix $\mathbf{H}^\top\mathbf{H}$. To characterize the geometry, we need the following definitions.

**Definition 1** (Neural Collapse (NC)). *NC occurs if* $\mathbf{h}_i = \mathbf{h}_j, \ \forall\ i, j : y_i = y_j$.

**Definition 2** ($k$-Orthogonal Frame ($k$-OF)). *We say that $k$ vectors $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_k] \in \mathbb{R}^{d\times k}$ form a $k$-OF if $\mathbf{V}^\top\mathbf{V} \propto \mathtt{I}_k$, i.e., for each pair of $(i, j) \in [k]$, $\|\mathbf{v}_i\| = \|\mathbf{v}_j\|$ and $\mathbf{v}_i^\top\mathbf{v}_j = 0$.*

When the within-class variation of embeddings is negligible, i.e., $\mathbf{H}$ satisfies NC, it suffices to focus on the class-mean embeddings $\mu_c = \frac{1}{n_c}\sum_{i:y_i=c}\mathbf{h}_i, c \in [k]$ and the respective matrices $\mathbf{M}_{d\times k} = [\mu_1, \cdots, \mu_k]$ and $\mathbf{G}_\mathbf{M} = \mathbf{M}^\top\mathbf{M}$ instead of $\mathbf{H}$ and $\mathbf{H}^\top\mathbf{H}$. With these, we can formally define the OF geometry for embeddings.

**Definition 3** (OF geometry). *We say that a feature-embedding matrix $\mathbf{H}$ follows an OF geometry if it satisfies NC and the class-means form a $k$-OF, i.e., $\mathbf{G}_\mathbf{M} \propto \mathtt{I}_k$.*

## 3 SCL WITH RELU LEARNS OF GEOMETRIES: EMPIRICAL FINDINGS

**Experimental setup.** Following the setup of Graf et al. (2021), our models consist of a backbone (ResNet, DenseNet, etc) with a normalizing layer to output features with unit norm. For all geometric convergence results, we apply a ReLU on output features before normalization. We employ Stochastic Gradient Descent (SGD) with a learning rate of 0.1, momentum set to 0.9 and with no weight decay. Furthermore, as in Graf et al. (2021); Khosla et al. (2020), we set the temperature parameter $\tau = 0.1$ as Khosla et al. (2020) have found that it yields optimal performance. In addition, we empirically find that convergence to OF is not highly dependent on $\tau$ for values near 0.1, yet the specific choice affects the speed of convergence.

We study the behavior of models trained with SCL under, 1) *R-STEP* imbalance having $k/2$ majority classes with $n_{\text{maj}}$ examples per class and $k/2$ minority classes with $n_{\text{min}} = n_{\text{maj}}/R$ examples per class, and 2) *R-Long-tailed* (LT) imbalance where the number of training datapoints exponentially decreases across classes such that $n_c = n_1 R^{-(c-1/k-1)}$, for $c \in [k]$. Regardless of the imbalance ratio $R$, we ensure that $n_c \geq 2$ by adding a vertically flipped version of each image as a method of batch duplication. Unless stated, we do not perform any additional data augmentation on the datasets, and use a batch size of 1024 with random reshuffling.

Finally, when studying the impact of ReLU on generalization, we use a Nearest Center Classifier (NCC) on the output features to evaluate model performance. For such experiments, following the setup of Khosla et al. (2020), we employ a projection head by adding a 2 layer non-linear MLP (with or without ReLU at the last layer) and compare the test accuracy under difference imbalance ratios.

**Metrics.** Since in all experiments we observe the within-class variations of the last-layer embeddings ($\mathbf{h}_i$) become negligible towards the end of training (see NC plots in Fig. 8 in SM), we focus here on the geometry of class-mean embeddings $\mathbf{M}$. Thus, to measure the distance of learned embedding to the OF geometry, we compute the distance metric $\mathtt{G}_\mathbf{M} := \left\| \frac{\mathbf{G}_\mathbf{M}}{\|\mathbf{G}_\mathbf{M}\|_F} - \frac{\mathtt{I}_k}{\|\mathtt{I}_k\|_F} \right\|_F$.

| Imbalance Ratio (R) | w/o ReLU | w/ ReLU |
|---|---|---|
| 1 | 72.17 ± 0.23 | 72.32 ± 0.60 |
| 10 (Step) | 56.58 ± 0.50 | 58.16 ± 0.76 |
| 10 (LT) | 57.10 ± 1.04 | 57.88 ± 1.05 |
| 100 (Step) | 43.49 ± 0.30 | 43.80 ± 0.25 |
| 100 (LT) | 37.19 ± 2.50 | 39.71 ± 0.09 |

**Table 1:** Test accuracy comparison for a ResNet-18 trained using SCL on CIFAR100 with and without ReLU after the projection head. We use NCC for classification. Note that the addition of ReLU does not compromise the accuracy. For details, see E.7 in SM.

**Observations on Geometry.** In Fig. 2, we plot the distance $\mathtt{G}_\mathbf{M}$ between the learned feature-embeddings and the OF geometry as training progresses. We consistently observe

[3] When using SCL, it is common practice to add an augmented version of the datapoints in each batch to itself (Khosla et al., 2020; Graf et al., 2021). This practice, referred to as *batch duplication*, helps ensure that each datapoint in a batch has at least one other example in the same class to compare to during training.

[4] The specific terminology is adopted from Behnia et al. (2023).

that the learned features during training converge to the OF geometry, irrespective of the imbalance level $R$ of the training set and the imbalance pattern (STEP or LT). This suggests that the feature geometry learned by SCL is invariant to the training label distribution.

Interestingly, the invariance that is revealed by our study is distinct for SCL with ReLU as opposed to the case of SCL without ReLU, and that of CE and MSE loss. CE and MSE are known to be sensitive to imbalances (Thrampoulidis et al., 2022; Fang et al., 2021; Liu et al., 2023; Behnia et al., 2023; Dang et al., 2023). Additionally, for different values of $R$, there is no significant difference between the speed of convergence, unlike the CE loss (Thrampoulidis et al., 2022), where it is empirically observed that the rate worsens with larger imbalance.

**Generalization.** To ensure that ReLU does not yield any adverse effects on test accuracy, we trained ResNet-18 (with a projection head) with and without ReLU on CIFAR10, CIFAR100, Tiny Imagenet and evaluated the balanced test accuracy when training under label imbalance. Following (Zhu et al., 2022; Khosla et al., 2020; Chen et al., 2020), we consider the features before a projection head for evaluating the test accuracy of the models. Our results, e.g. in Tab. 1, indicate that the addition of ReLU does *not* compromise test accuracy. For a detailed discussion on the results see E.7 in SM.

## 4 Theoretical Justification: SCL with Non-Negativity Constraints

In this section, we analytically justify the convergence of the embeddings learned by SCL to the OF geometry. For our theoretical analysis, we use the Unconstrained Features Model (UFM) (Mixon et al., 2020; Fang et al., 2021; Graf et al., 2021; Ji et al., 2021; Lu & Steinerberger, 2020; Tirer & Bruna, 2022), where we treat the last-layer features $\mathbf{H}$ as free variables, removing the dependence to the network parameters . However, we refine the UFM, and consider the SCL minimization over the non-negative orthant to accommodate for the presence of ReLU activations in the last layer.[5] Following Khosla et al. (2020), we further constrain the embeddings $\mathbf{h}_i$ to be normalized. We prove, irrespective of the labels distribution, that the global optimizers form an OF. Formally, consider the following refined UFM for SCL, which we call $\text{UFM}_+$ for convenience:

$$\widehat{\mathbf{H}} \in \arg\min_{\mathbf{H}} \mathcal{L}_{\text{SCL}}(\mathbf{H}) \;\; \text{subj. to} \;\; \mathbf{H} \geq 0 \;\; \text{and} \;\; \|\mathbf{h}_i\|^2 = 1, \; \forall i \in [n]. \tag{UFM$_+$}$$

Recall $\mathbf{H} \geq 0$ denotes entry-wise non-negativity. We begin our analysis by considering the full-batch loss in Sec. 4.1, and extend our results to the SCL minimized on mini-batches in Sec. 4.2. We provide specific conditions that mini-batches must satisfy in order to have the same global optimum as the full-batch version.

### 4.1 Full-batch SCL

Here, we focus on the full-batch SCL, where the entire training set is treated as a single batch. Specifically, we consider $\text{UFM}_+$ with $\mathcal{L}_{\text{SCL}}(\mathbf{H})$ being the full-batch SCL defined as,

$$\mathcal{L}_{\text{full}}(\mathbf{H}) := \sum_{i\in[n]} \frac{1}{n_{y_i} - 1} \sum_{j\neq i; y_j = y_i} \log\Big( \sum_{\ell \neq i} \exp\left(\mathbf{h}_i^\top \mathbf{h}_\ell - \mathbf{h}_i^\top \mathbf{h}_j\right)\Big). \tag{2}$$

Thm. 1 specifies the optimal cost and optimizers of $\text{UFM}_+$ with the full-batch SCL (2).

**Theorem 1** (Full-batch SCL minimizers)**.** *Let $d \geq k$. For any $\mathbf{H}$ feasible in $\text{UFM}_+$, it holds,*

$$\mathcal{L}_{\text{full}}(\mathbf{H}) \geq \sum_{c\in[k]} n_c \log\left(n_c - 1 + (n - n_c)e^{-1}\right). \tag{3}$$

*Moreover, equality is achieved if and only if $\mathbf{H}$ satisfies NC and the class-means form a k-OF.*

We defer the proof details to Sec. A in SM. The bound relies on successive uses of Jensen's inequality and the fact that for feasible $\mathbf{H}$, each pair of training samples $i, j \in [n]$ satisfies $0 \leq \mathbf{h}_i^\top \mathbf{h}_j \leq 1$. We complete the proof by verifying that equality is attained only if $\mathbf{h}_i^\top \mathbf{h}_j = 1$ when $y_i = y_j$ and $\mathbf{h}_i^\top \mathbf{h}_j = 0$ when $y_i \neq y_j$. Rather, to achieve the optimal cost, features with similar labels must align (NC) and the class-mean features must form an OF.

[5] Tirer & Bruna (2022) (and several follow ups, e.g. Súkeník et al. (2023)) has also studied incorporating the ReLU activation in the UFM, albeit focusing on MSE loss.

Thm. 1 shows that any optimal embedding geometry learned by the full-batch SCL with ReLU constraints uniquely follows the OF geometry (Defn. 3) and the conclusion is independent of the training label distribution. We have already seen in Sec. 3, that this conclusion is empirically verified by deep-net experiments. To further verify the lower bound on the cost of the SCL loss given by the theorem, we compare it in Fig. 3 with the empirical loss of a ResNet-18 model trained with full-batch SCL (2) on $R$-STEP MNIST dataset and $n = 1000$ total examples. Note the remarkable convergence of the loss to the lower bound (dashed horizontal lines) as training progresses.

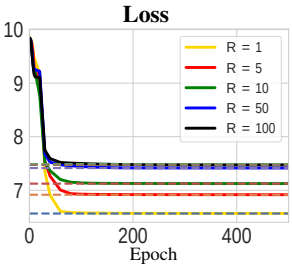



**Figure 3:** Full-batch SCL converges to the lower bound (dashed lines) in Thm. 1.

### 4.2 Mini-batch SCL

Thm. 1 shows that minimizing SCL over the training set as a single batch recovers an OF in the feature space, regardless of the training label distribution. However, in practice, SCL optimization is performed over batches chosen from training set (Khosla et al., 2020; Graf et al., 2021). Specifically, we have a set of batches $\mathcal{B}$ and we compute the loss on each batch $B \in \mathcal{B}$ as in (1). While in the full-batch SCL (2), all pairs of training samples interact with each other, in the mini-batch version, two samples $(i, j)$ interact only if there exists a batch $B \in \mathcal{B}$ such that $i, j \in B$. A natural question is whether the mini-batch construction impacts the embeddings learned. In this section, we study the role of batching on the embeddings geometry more closely.

Similar to the previous section, we consider UFM$_+$, where we directly optimize the SCL over embeddings $\mathbf{H}$. However, this time we study the mini-batch SCL defined as follows,

$$\mathcal{L}_{\text{batch}}(\mathbf{H}) := \sum_{B\in\mathcal{B}} \sum_{i\in B} \frac{1}{n_{B;y_i} - 1} \sum_{\substack{j\in B \\ j\neq i; y_j = y_i}} \log\left( \sum_{\substack{\ell\in B \\ \ell\neq i}} \exp\left(\mathbf{h}_i^\top \mathbf{h}_\ell - \mathbf{h}_i^\top \mathbf{h}_j\right)\right), \tag{4}$$

where recall $n_{B;c} = |\{i : i \in B, y_i = c\}|$. Thm. 2 is an extension of Thm. 1 for the mini-batch SCL (4). The proof proceeds similarly to that of Thm. 1, where the loss over each batch can be independently bounded from below. Interestingly, due to the orthogonality of the optimal embeddings in every batch, the lower bound for each batch can be achieved by the common minimizer. The detailed proof is deferred to Sec. B in the SM.

**Theorem 2** (Mini-batch SCL minimizers)**.** *Let $d \geq k$ and $\mathcal{B}$ be an arbitrary set of batches of examples. For any feasible $\mathbf{H}$ in UFM$_+$, the following lower bound holds,*

$$\mathcal{L}_{\text{batch}}(\mathbf{H}) \geq \sum_{B\in\mathcal{B}} \sum_{c\in[k]} n_{B;c} \log\left(n_{B;c} - 1 + (n_B - n_{B;c})e^{-1}\right). \tag{5}$$

*Equality holds if and only if for every $B \in \mathcal{B}$ and every pair of samples $i, j \in B$, $\mathbf{h}_i^\top \mathbf{h}_j = 0$ if $y_i \neq y_j$ and $\mathbf{h}_i = \mathbf{h}_j$ if $y_i = y_j$.*

We remark that the only previous study of SCL geometry with batches by Graf et al. (2021) requires significantly more restrictive conditions on the batch set $\mathcal{B}$ to guarantee a common geometry among all batches $B \in \mathcal{B}$. Specifically, Graf et al. (2021) assumes a batch set that includes all possible combinations of examples. Instead, thanks to the addition of ReLU, our requirements are significantly relaxed. This is detailed in the following remark. See also Sec. B.2 in SM.

**Remark 1** (Comparison to analysis of Graf et al. (2021))**.** *We elaborate on the comparison to Graf et al. (2021) with an illustrative example below. When optimizing the loss decomposed over a set of mini-batches, it is crucial to consider whether the individual batch minimizers are also overall optimal solutions. Consider the following setting: $k = d = 3, y_1 = 1, y_2 = 2, y_3 = 3, B_1 = \{1, 2\}, B_2 = \{1, 3\}, B_3 = \{2, 3\}, \mathcal{B} = \{B_1, B_2, B_3\}$. Let us identify the optimal embeddings for the two scenarios: (i) with and (ii) without non-negativity constraints on the embedding coordinates.*
***(i) Without non-negativity (UFM)**, the optimal embedding configurations can be shown to be ETF with 2 vectors for every batch, which is simply an antipodal structure. In other words, the batch-wise optimal solutions are $\mathbf{h}_1 = -\mathbf{h}_2, \mathbf{h}_1 = -\mathbf{h}_3$ and $\mathbf{h}_2 = -\mathbf{h}_3$, for the batches $B_1, B_2$ and $B_3$, respectively. However, the batch-wise optimal solutions are infeasible due to their contradictory nature. From Graf et al. (2021), it can be deduced that the overall optimal configuration is instead an ETF with 3 vectors. This example underscores the difficulty in optimizing the loss over every batch separately in*

*case of SCL without non-negativity. See (a) in Fig. 6 in SM.*
***(ii) With non-negativity*** *(UFM$_+$), our results imply that the optimal embeddings form a 2-OF for every batch, i.e., $\mathbf{h}_1 \perp \mathbf{h}_2, \mathbf{h}_1 \perp \mathbf{h}_3$ and $\mathbf{h}_2 \perp \mathbf{h}_3$, for the batches $B_1, B_2$ and $B_3$, respectively. The three conditions are compatible with each other and the fact that the overall optimal solution is a 3-OF in $\mathbb{R}^3$. Therefore, we were able to break down the overall optimization into individual batches. See (b) in Fig. 6 in SM.*

**Global minimizer geometry may not be unique.** It is easy to verify that any $\mathbf{H}$ following the OF geometry achieves the lower-bound of Thm. 2 and it is also a global optimizer of the mini-batch SCL. However, depending on the choice of $\mathcal{B}$, the lower bound can possibly be attained by other optimal embeddings that do not necessarily satisfy NC or orthogonality. Hence the global optimizer may not have a unique geometry.

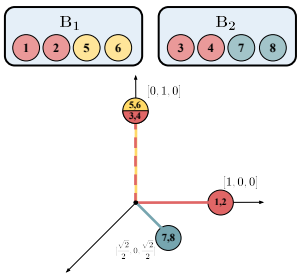



**Figure 4:** A non-OF geometry minimizing mini-batch SCL with $\mathcal{B} = \{B_1, B_2\}$. Samples with different labels are marked with different colors.

To highlight the importance of $\mathcal{B}$ in the characterization of the optimal geometry of UFM$_+$ when using the batch-loss, consider the toy example in Fig. 4. Suppose we have $k = 3$ classes and we want to find the optimal normalized and non-negative features in $\mathbb{R}^3$ by minimizing (4) for $\mathcal{B} = \{B_1, B_2\}$. Although the OF satisfies the global optimality condition in Thm. 2, the theorem requires milder conditions for the optimal $\widehat{\mathbf{H}}$: $\widehat{\mathbf{h}}_i$ and $\widehat{\mathbf{h}}_j$ need to be aligned ($y_i = y_j$) or orthogonal ($y_i \neq y_j$) only if they interact within one of the selected batches. Fig. 4 shows one such optimal geometry that does not satisfy either of NC or orthogonality conditions: Firstly, the samples $i = 1$ and $i = 3$ have significantly different features despite belonging to the same class. Second, $\widehat{\mathbf{h}}_3, \widehat{\mathbf{h}}_4$ align with $\widehat{\mathbf{h}}_5, \widehat{\mathbf{h}}_6$ although they have different labels, and $\widehat{\mathbf{h}}_7, \widehat{\mathbf{h}}_8$ are not orthogonal to samples $\widehat{\mathbf{h}}_1, \widehat{\mathbf{h}}_2$. With this example, we are now ready to discuss the role of batching more formally in the next section.

## 5 Batching Matters

Since the OF may not be the only solution for the UFM$_+$ with mini-batch SCL (4), here we study the role of mini-batches to avoid ambiguous solutions with different geometries. In Sec. 5.1, we identify necessary and sufficient conditions for a batching strategy to yield a unique global solution geometry. By this result, in Sec. 5.2, we propose a simple yet effective scheme that provably succeeds in improving the convergence to an OF.

### 5.1 When is OF geometry the unique minimizer?

As discussed in Sec. 4.2 the uniqueness of the global minimizer's geometry when considering mini-batches depends on the interaction of samples in the loss function, or, in other words, the choice of $\mathcal{B}$. Before specifying for which $\mathcal{B}$ the minimizer is unique, we need to define the Batch Interaction Graph, a graph that captures the pairwise interactions of samples within the batches.

**Definition 4** (Batch Interaction Graph)**.** *Consider an undirected graph $G = (V, E)$ where $V := [n]$. We define the Batch Interaction Graph for a given set of batches $\mathcal{B}$ as follows: vertices $u, v \in V$ are connected if and only if there exists a batch $B \in \mathcal{B}$ such that $u, v \in B$. Moreover, $G_c$ denotes the induced subgraph of $G$ with vertices $V_c = \{u : y_u = c\}$.*

We state necessary and sufficient conditions on $\mathcal{B}$ for the minimizer of mini-batch SCL to be unique:

**Corollary 2.1.** *Consider the Batch Interaction Graph $G$ corresponding to $\mathcal{B}$. The global optimizer geometry of UFM$_+$ with mini-batch SCL (4) is unique and forms an OF if and only if $G$ satisfies the following conditions: (1) For every class $c \in [k]$, $G_c$ is a connected graph. (2) For every pair $c_1, c_2 \in [k]$, there exists at least one edge between $G_{c_1}$ and $G_{c_2}$.*

Corollary 2.1 serves as a check for whether a given batching scheme yields a unique global minimizer geometry or not. It also provides guidance for designing mini-batches. We elaborate on this below.

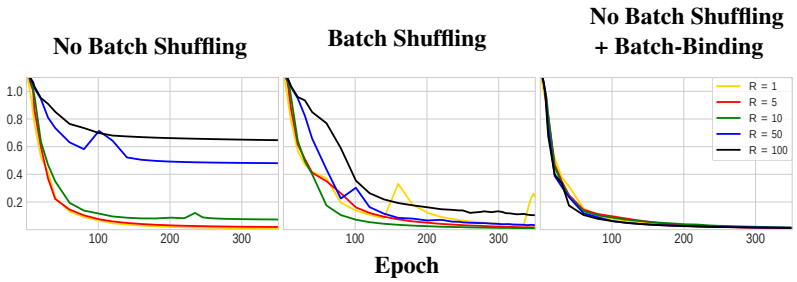



**Figure 5:** Convergence to the OF geometry for various batching schemes including the analysis-inspired scheme "No Batch Shuffling + Batch-Binding". ResNet-18 trained on CIFAR-10 with a small batch size of 128, see text. See also Fig. 18 in SM

### 5.2 *Batch-binding* ENSURES UNIQUE OF MINIMIZER AND IMPROVES CONVERGENCE

Corollary 2.1 shows that an arbitrary set of mini-batches is not guaranteed to have OF geometry as its minimizing configuration in the embedding space. To enforce the OF geometry as the unique minimizer, we introduce a simple scheme with low computational overhead as follows. We define **binding examples** as *a set of examples including exactly one example from every class*. This set has, thus, $k$ elements. The process of **batch-binding** is simply adding the above set of binding examples to every mini-batch in the given original set. See Sec. D of SM for an algorithm and Fig. 16 in SM for a visual example.

Batch-binding adds a constant $k$ to the size of every mini-batch. $k$ is typically smaller than the batch sizes used to train SCL which could range from 2048 to 6144 (Khosla et al., 2020). While adding small computational overhead, this method guarantees that $\mathcal{B}'$ satisfies the conditions of Cor. 2.1 and SCL learns an OF geometry. We conduct a series of experiments to illustrate the impact of batch binding on convergence to OF.

To verify the role of batch selection on the learned features, we train ResNet-18 on CIFAR-10 under three batching scenarios: 1) **No batch-shuffling**: a mini-batch set $\mathcal{B}$ that partitions the training examples once at initialization and is held constant across training epochs. 2) **Batch-shuffling**: the examples are divided into mini-batch partition, with a random reshuffle of the examples at *every* epoch. 3) **No batch-shuffling + batch-binding**: we construct a fixed partition of examples into mini-batches; then, we perform batch-binding. In order to focus on the impact of batch selection strategies, we consider a relatively small batch-size of 128.

We remark that Thm. 2 and Cor. 2.1 can explain the behaviors of all three schemes. Moreover, the batch-binding strategy guided by our analytical results is effective at ensuring fast and predictable convergence of deep-nets to a unique OF geometry. Fig. 5 shows the distance of learned embeddings to the OF geometry for the three batching scenarios mentioned above. In case of the fixed partition batching (left), the features do not converge to OF, especially with large imbalance ratios. This behavior is consistent with the conditions identified by Thm. 2, because in a typical random partition, the corresponding induced subgraphs are not necessarily connected. On the other hand, when the batching is performed with a randomly reshuffled data ordering (center), the geometry converges more closely to the OF, albeit with certain epochs deviating significantly from the convergence direction. We hypothesize that random reshuffling creates an opportunity for different examples to interact sufficiently, thus, when trained long enough, to converge close to an OF geometry. Finally, with the third batching strategy of batch-binding (right), even without shuffling the samples at every epoch, we observe a consistently fast convergence to the global optimizer, OF. These observations provide strong evidence supporting our analysis of batching predicting the behavior of SCL trained deep-nets.

While the above studies the impact of batching itself, we perform a number of additional experiments in order to illustrate the impact of binding examples. In particular, we observe that batch binding helps improve convergence to OF geometry when training with less powerful models (see Fig. 19 for DenseNet experiments in SM), more complex dataset (see Fig. 20 in SM and Fig. 2 for CIFAR100 and TinyImageNet results) and even convergence to ETF in the absence of ReLU under balanced training (See Fig. 21 in SM). Such results emphasize the importance of batch binding and encourage further analysis of batching under contrastive learning.

## 6 More Related Work

The simple concept behind SCL, introduced as an extension of the contrastive loss (e.g., Chen et al., 2020; Tian et al., 2019) to the fully supervised setting by Khosla et al. (2020), involves pulling together the normalized features of examples belonging to the same class while pushing them away from examples of other classes. Despite its simplicity, SCL offers a generalization of existing loss functions like the triplet loss (Weinberger et al., 2005) and the N-pair loss (Sohn, 2016), and surpasses the performance of the CE loss on standard vision classification datasets, while also being more robust to natural corruptions in the data and less sensitive to hyperparameters (Khosla et al., 2020). The feature geometry of the self-supervised constrastive loss was considered by (Wang & Isola, 2020), where the loss function was shown to optimize *uniformity* and *alignment* of features, as the number of negative examples approaches infinity. Instead, we study the supervised loss, providing exact non-asymptotic solutions for the corresponding UFM and the impact of batching in detail.

In recent years, there has been a notable interest regarding the properties of the embedding space and weights learned through unsupervised/supervised contrastive losses, as well as CE. Of particular importance is the endeavor to uncover the fundamental distinctions between these losses to effectively harness each of their strengths and formalize principled ways to combine them. Specifically, starting by Papyan et al. (2020), an increasing series of recent works have focused on uncovering the embedding geometry of CE, Mean-Squared Error (MSE) losses, and their variants. We highlighted some of these works in Sec. 1 while numerous other works contribute to these investigation (e.g., Zhou et al., 2022a;b; Yaras et al., 2022; Gao et al., 2023; Han et al., 2021; Súkeník et al., 2023). In addition to the scientific curiosity surrounding such studies, this exploration has paved the way for novel CE-based approaches to training DNNs on imbalanced data (Behnia et al., 2023; Liang & Davis, 2023; Xie et al., 2023; Sharma et al., 2023; Dang et al., 2023; Yang et al., 2022).

Less attention has been devoted in the existing literature to a comparable set of results for the SCL. Graf et al. (2021) is the first to study the geometry of SCL for balanced data and comparing it to CE. To tackle imbalances, and inspired by Graf et al. (2021) that suggests balanced data is crucial for obtaining symmetric embeddings, Zhu et al. (2022) have recently proposed and investigated a modification of the SCL that boosts performance under class imbalances. Our work extends and complements the two studies by showing that SCL with ReLU can actually learn symmetric embedding structure even in the presence of imbalances. While preparing the manuscript, we became aware of a recent work (Cho et al., 2023), where the authors consider the mini-batch optimization of the unsupervised contrastive loss and propose a choice of mini-batches to speed up convergence.

## 7 Concluding Remarks

We have shown consistent empirical and theoretical evidence that SCL learns training embeddings that converge to the OF geometry irrespective of the level of class imbalance. We believe this finding contributes a unique result to the growing literature of neural-collapse / implicit-geometry phenomena. For balanced data, we extend the contributions of Graf et al. (2021) to the case in presence of ReLU activations. Further, our results identify exact conditions on the mini-batch selection to achieve the OF geometry, allowing for a wider set of possibilities in batching than found by Graf et al. (2021) for SCL without ReLU. For imbalanced data, our results are the first explicit characterization of the geometry, concluding that imbalances do *not* alter the geometry if a ReLU activation is added at the last layer. In both cases, these advancements are achieved by analyzing a refined UFM model, $\text{UFM}_+$, that closely aligns with experimental conditions by constraining embeddings to the non-negative orthant. This model also leads to new findings about the intricate role of batching in SCL optimization, that may be of independent interest. A future investigation of exact characterization of the geometry with imbalances and without ReLU can help complete the understanding of the implicit-geometry of SCL. Although the majority of implicit geometry characterizations in the literature require $d > k$, it is interesting to extend our findings to settings where $d < k$ following the steps of Gao et al. (2023) for the CE loss. Likewise, our finding regarding the crucial role of batching in geometry opens the door to further investigations aimed at devising efficient batch strategies in cases with a large number of classes. Finally, like most studies on neural-collapse phenomena, our results share a common limitation: they only provide insights into the behavior during training. On the other hand, there is a line of research that explores algorithmic modifications to SCL (Gunel et al., 2020; Jitkrittum et al., 2022; Samuel & Chechik, 2021; Li et al., 2022; Kang et al., 2021; Li et al., 2022), often rooted but not explicitly connected to geometric principles, to enhance generalization under imbalances. Ultimately, we envision the two research streams merging through the ongoing exchange of ideas.

## REFERENCES


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*, pp. 10815–10838. PMLR, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Jaewoong Cho, Kartik Sreenivasan, Keon Lee, Kyunghoo Mun, Soheun Yi, Jeong-Gwan Lee, Anna Lee, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Mini-batch optimization of contrastive loss. *arXiv preprint arXiv:2307.05906*, 2023.

Hien Dang, Tan Nguyen, Tho Tran, Hung Tran, and Nhat Ho. Neural collapse in deep linear network: From balanced to imbalanced data. *arXiv preprint arXiv:2301.00437*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

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.

Peifeng Gao, Qianqian Xu, Peisong Wen, Huiyang Shao, Zhiyong Yang, and Qingming Huang. A study of neural collapse phenomenon: Grassmannian frame, symmetry, generalization. *arXiv preprint arXiv:2304.08914*, 2023.

Florian Graf, Christoph Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised constrastive learning. In *International Conference on Machine Learning*, pp. 3821–3830. PMLR, 2021.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*, 2020.

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.

Wenlong Ji, Yiping Lu, Yiliang Zhang, Zhun Deng, and Weijie J Su. An unconstrained layer-peeled perspective on neural collapse. *arXiv preprint arXiv:2110.02796*, 2021.

Wittawat Jitkrittum, Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Elm: Embedding and logit margins for long-tail learning. *arXiv preprint arXiv:2204.13208*, 2022.

Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio S Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6918–6928, 2022.

Tong Liang and Jim Davis. Inducing neural collapse to a fixed hierarchy-aware frame for reducing mistake severity. *arXiv preprint arXiv:2303.05689*, 2023.

Xuantong Liu, Jianfeng Zhang, Tianyang Hu, He Cao, Yuan Yao, and Lujia Pan. Inducing neural collapse in deep long-tailed learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 11534–11544. PMLR, 2023.

Jianfeng Lu and Stefan Steinerberger. Neural collapse with cross-entropy loss. *arXiv preprint arXiv:2012.08465*, 2020.

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

Vardan Papyan. The full spectrum of deep net hessians at scale: Dynamics with sample size. *arXiv preprint arXiv:1811.07062*, 2018.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Dvir Samuel and Gal Chechik. Distributional robustness loss for long-tail learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9495–9504, 2021.

Saurabh Sharma, Yongqin Xian, Ning Yu, and Ambuj Singh. Learning prototype classifiers for long-tailed recognition. *arXiv preprint arXiv:2302.00491*, 2023.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

Peter Súkeník, Marco Mondelli, and Christoph Lampert. Deep neural collapse is provably optimal for the deep unconstrained features model. *arXiv preprint arXiv:2305.13165*, 2023.

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

Y Tian, D Krishnan, and P Isola. Contrastive multiview coding. arxiv. *arXiv preprint arXiv:1906.05849*, 2019.

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

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, 18, 2005.

Liang Xie, Yibo Yang, Deng Cai, and Xiaofei He. Neural collapse inspired attraction-repulsion-balanced loss for imbalanced learning. *Neurocomputing*, 2023.

Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? *Advances in Neural Information Processing Systems*, 35:37991–38002, 2022.

Can Yaras, Peng Wang, Zhihui Zhu, Laura Balzano, and Qing Qu. Neural collapse with normalized features: A geometric analysis over the riemannian manifold. *arXiv preprint arXiv:2209.09211*, 2022.

Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. *arXiv preprint arXiv:2203.01238*, 2022a.

Jinxin Zhou, Chong You, Xiao Li, Kangning Liu, Sheng Liu, Qing Qu, and Zhihui Zhu. Are all losses created equal: A neural collapse perspective. *arXiv preprint arXiv:2210.02192*, 2022b.

Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6908–6917, 2022.

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.

## CONTENTS

**Additional Notations.** While recalling the notation mentioned in Sec. 1, we note a few more below. $\otimes$ denotes Kronecker products. We use $\mathbf{1}_m$ to denote an $m$-dimensional vector of all ones. For vectors/matrices with all zero entries, we simply write $0$, as dimensions are easily understood from context. $\mathbf{V}^\dagger$ its Moore-Penrose pseudoinverse. $\nabla_{\mathbf{V}}\mathcal{L} \in \mathbb{R}^{m\times n}$ is the gradient of a scalar differentiable function $\mathcal{L}(.)$ with respect to $\mathbf{V}$.

**List of Frequently Used Abbreviations.**

| | |
|---|---|
| CE | Cross Entropy |
| ETF | Equiangular Tight Frame |
| LT | Long-Tailed |
| MLP | Multi-Layer Perceptron |
| MSE | Mean Squared Error |
| NC | Neural Collapse |
| NCC | Nearest Class-Center |
| OF | Orthogonal Frame |
| ReLU | Rectified Linear Unit |
| SCL | Supervised Contrastive Loss |
| UFM | Unconstrained Features Model |

## A Proof of Thm. 1

For $c \in [k]$, let $\mathcal{C}_c = \{i : y_i = c\}$ be the set of examples belonging to class $c$, and define $\mathcal{L}_i(\mathbf{H})$ as follows,

$$\begin{aligned}\mathcal{L}_i(\mathbf{H}) &= \sum_{j\in\mathcal{C}_{y_i}; j\neq i} \log\Big(\sum_{\ell\in[n]; \ell\neq i} \exp\left(\mathbf{h}_i^\top\mathbf{h}_\ell - \mathbf{h}_i^\top\mathbf{h}_j\right)\Big)\\ &= \sum_{j\in\mathcal{C}_{y_i}; j\neq i} \log\Big(\sum_{\ell\in\mathcal{C}_{y_i}; \ell\neq i} \exp\left(\mathbf{h}_i^\top\mathbf{h}_\ell - \mathbf{h}_i^\top\mathbf{h}_j\right) + \sum_{\ell\notin\mathcal{C}_{y_i}} \exp\left(\mathbf{h}_i^\top\mathbf{h}_\ell - \mathbf{h}_i^\top\mathbf{h}_j\right)\Big). \qquad (6)\end{aligned}$$

Then, we can rewrite the full-batch SCL in (2) as,

$$\mathcal{L}_{\text{full}}(\mathbf{H}) = \sum_{i\in[n]} \frac{1}{n_{y_i}-1}\mathcal{L}_i(\mathbf{H}) = \sum_{c\in[k]} \frac{1}{n_c-1} \sum_{i\in\mathcal{C}_c} \mathcal{L}_i(\mathbf{H}).$$

Now for $c \in [k]$, consider example $i \in \mathcal{C}_c$, and define

$$\mathbf{a}_i := \frac{1}{n_c-1} \sum_{\grave{}\in\mathcal{C}_c;\grave{}\neq i} \mathbf{h}_{\grave{}}, \quad \mathbf{b}_i := \frac{1}{n-n_c} \sum_{\grave{}\notin\mathcal{C}_c} \mathbf{h}_{\grave{}}.$$

Note $\mathbf{a}_i$ is the mean-embedding of all examples in class $c$ but $i$, and $\mathbf{b}_i$ is the mean-embedding of all examples not in class $c$. Starting from (6), by applying Jensen's inequality to the strictly convex function $f_1(x) := \exp(x - \mathbf{h}_i^\top \mathbf{h}_j)$ we have,

$$\begin{aligned}
\mathcal{L}_i(\mathbf{H}) &\geq \sum_{j\in\mathcal{C}_c;j\neq i} \log\Big((n_c-1)\exp\left(\mathbf{h}_i^\top\mathbf{a}_i - \mathbf{h}_i^\top\mathbf{h}_j\right) + (n-n_c)\exp\left(\mathbf{h}_i^\top\mathbf{b}_i - \mathbf{h}_i^\top\mathbf{h}_j\right)\Big) \\
&= \sum_{j\in\mathcal{C}_c;j\neq i} \log\Big(\big((n_c-1)\exp\left(\mathbf{h}_i^\top\mathbf{a}_i\right) + (n-n_c)\exp\left(\mathbf{h}_i^\top\mathbf{b}_i\right)\big)\exp\left(-\mathbf{h}_i^\top\mathbf{h}_j\right)\Big) \\
&\overset{(i)}{=} \sum_{j\in\mathcal{C}_c;j\neq i} \log\Big(\alpha_i \exp\left(-\mathbf{h}_i^\top\mathbf{h}_j\right)\Big) \\
&\overset{(ii)}{=} (n_c-1)\log\Big(\alpha_i \exp\big(-\mathbf{h}_i^\top\big(\frac{1}{n_c-1}\sum_{j\in\mathcal{C}_c;j\neq i}\mathbf{h}_j\big)\big)\Big) \\
&= (n_c-1)\log(\alpha_i\exp(-\mathbf{h}_i^\top\mathbf{a}_i)) \\
&= (n_c-1)\log\Big((n_c-1) + (n-n_c)\exp\big(\frac{1}{n-n_c}\mathbf{h}_i^\top\sum_{\grave{}\notin\mathcal{C}_c}\mathbf{h}_{\grave{}} - \frac{1}{n_c-1}\mathbf{h}_i^\top\sum_{\grave{}\in\mathcal{C}_c;\grave{}\neq i}\mathbf{h}_{\grave{}}\big)\Big),
\end{aligned} \tag{7}$$

where in $(i)$, we define $\alpha_i := (n_c-1)\exp\left(\mathbf{h}_i^\top\mathbf{a}_i\right) + (n-n_c)\exp\left(\mathbf{h}_i^\top\mathbf{b}_i\right)$, and in $(ii)$, we use the fact that the function $f_2(x) := \log(\alpha_i\exp(x))$ is an affine function. Now consider all samples $i \in \mathcal{C}_c$. From (7),

$$\begin{aligned}
\frac{1}{n_c-1}\sum_{i\in\mathcal{C}_c}\mathcal{L}_i(\mathbf{H}) &\geq \sum_{i\in\mathcal{C}_c}\log\Big((n_c-1) + (n-n_c)\exp\big(\frac{1}{n-n_c}\mathbf{h}_i^\top\sum_{\grave{}\notin\mathcal{C}_c}\mathbf{h}_{\grave{}} - \frac{1}{n_c-1}\mathbf{h}_i^\top\sum_{\grave{}\in\mathcal{C}_c;\grave{}\neq i}\mathbf{h}_{\grave{}}\big)\Big) \\
&\geq n_c\log\Big((n_c-1) + (n-n_c)\exp\big(\frac{1}{n_c(n-n_c)}\sum_{\substack{i\in\mathcal{C}_c\\ \grave{}\notin\mathcal{C}_c}}\mathbf{h}_i^\top\mathbf{h}_{\grave{}} - \frac{1}{n_c(n_c-1)}\sum_{\substack{i\in\mathcal{C}_c\\ \grave{}\in\mathcal{C}_c;\grave{}\neq i}}\mathbf{h}_i^\top\mathbf{h}_{\grave{}}\big)\Big).
\end{aligned} \tag{8}$$

In the last line we use Jensen's inequality on $f_3(x) := \log((n_c-1) + (n-n_c)\exp(x))$ which is strictly convex since $n_c > 1$ and $n - n_c > 0$.

By the ReLU constraints we have $\mathbf{h}_i \geq 0$, $i \in [n]$, which implies $\mathbf{h}_i^\top\mathbf{h}_j \geq 0$, $\forall i,j \in [n]$. Further, by Cauchy-Schwarz inequality $\mathbf{h}_i^\top\mathbf{h}_j \leq \|\mathbf{h}_i\|\|\mathbf{h}_j\| = 1$, with equality achieved if and only if $\mathbf{h}_i = \mathbf{h}_j$. Since $f_3$ is non-decreasing, we can simplify the bound in (8) as follows,

$$\frac{1}{n_c-1}\sum_{i\in\mathcal{C}_c}\mathcal{L}_i(\mathbf{H}) \geq n_c\log\Big((n_c-1) + (n-n_c)e^{-1}\Big). \tag{9}$$

Summing over all classes $c \in [k]$, we get the final bound on the full-batch SCL:

$$\mathcal{L}_{\text{full}}(\mathbf{H}) = \sum_{c\in[k]}\frac{1}{n_c-1}\sum_{i\in\mathcal{C}_c}\mathcal{L}_i(\mathbf{H}) \geq \sum_{c\in[k]} n_c\log\Big((n_c-1) + (n-n_c)e^{-1}\Big).$$

To achieve the lower-bound, from (9), we require $\mathbf{h}_i = \mathbf{h}_j$ if $y_i = y_j$ and $\mathbf{h}_i^\top\mathbf{h}_j = 0$ otherwise. This requirement also satisfies the equality condition for the Jensen inequalities applied in (7) and (8). Thus, any $\mathbf{H}$ achieving the lower-bound follows an OF geometry (Def. 3), which exists as long as $d \geq k$.

## B PROOF OF THM. 2

Consider the mini-batch SCL in (4), repeated below for convenience:

$$\mathcal{L}_{\text{batch}}(\mathbf{H}) := \sum_{B\in\mathcal{B}}\sum_{i\in B}\frac{1}{n_{B;y_i}-1}\sum_{\substack{j\in B\\ j\neq i; y_j=y_i}}\log\left(\sum_{\substack{\ell\in B\\ \ell\neq i}}\exp\left(\mathbf{h}_i^\top\mathbf{h}_\ell-\mathbf{h}_i^\top\mathbf{h}_j\right)\right),$$

Denoting the loss over a batch $B$ by

$$\mathcal{L}_B(\mathbf{H}) := \sum_{i\in B}\frac{1}{n_{B;y_i}-1}\sum_{\substack{j\in B\\ j\neq i; y_j=y_i}}\log\left(\sum_{\substack{\ell\in B\\ \ell\neq i}}\exp\left(\mathbf{h}_i^\top\mathbf{h}_\ell-\mathbf{h}_i^\top\mathbf{h}_j\right)\right),$$

we have $\mathcal{L}_{\text{batch}}(\mathbf{H}) = \sum_{B\in\mathcal{B}}\mathcal{L}_B(\mathbf{H})$. Now, for a given batch $B$, we apply Thm. 1 to get the lower bound and conditions for equality:

$$\mathcal{L}_B(\mathbf{H}) \geq \sum_{c\in[k]} n_{B;c}\log\left(n_{B;c}-1+(n_B-n_{B;c})e^{-1}\right).$$

Moreover, equality holds if and only if for every pair of samples $i,j\in B$: $\mathbf{h}_i^\top\mathbf{h}_j = 0$ if $y_i\neq y_j$ and $\mathbf{h}_i = \mathbf{h}_j$ if $y_i = y_j$. With this, the overall loss can be bounded from below by summing the individual lower bounds over different batches:

$$\mathcal{L}_{\text{batch}}(\mathbf{H}) \geq \sum_{B\in\mathcal{B}}\sum_{c\in[k]} n_{B;c}\log\left(n_{B;c}-1+(n_B-n_{B;c})e^{-1}\right),$$

with equality achieved if and only if for every $B\in\mathcal{B}$ and every pair of samples $i,j\in B$, $\mathbf{h}_i^\top\mathbf{h}_j = 0$ if $y_i\neq y_j$ and $\mathbf{h}_i = \mathbf{h}_j$ if $y_i = y_j$. As long as $d\geq k$, the equality conditions of every individual batch can be satisfied simultaneously.

### B.1 PROOF OF COR. 2.1

From Thm. 2, the optimal configuration of embeddings in UFM$_+$ under the mini-batch SCL depends on how batch conditions interact with each other. Specifically, recall that the equality in (5) is achieved if and only if for any batch $B\in\mathcal{B}$, and each pair of samples $(i,j)\in B$, $\mathbf{h}_i = \mathbf{h}_j$ if $y_i = y_j$, and $\mathbf{h}_i^\top\mathbf{h}_j = 0$ otherwise. The OF geometry clearly satisfies all these conditions and achieves the minimal cost of UFM$_+$. However, as discussed in Sec. 4.2, these equality conditions may be satisfied by configurations other than OF for an arbitrary batching scheme. So, in Cor. 2.1, we specify the requirements of a batching scheme in order for the global optimal of the mini-batch SCL to be unique (up to global rotations), and match the optimal of the full-batch SCL, which is the OF geometry.

To prove the corollary, we separately address the 'IF' and 'ONLY IF' parts of the Cor. 2.1.

• **'IF' direction.** Assume the batching scheme satisfies both conditions of Cor. 2.1: 1) $\forall c\in[k]$, the induced subgraph $G_c$ is connected, and 2) $\forall c\neq c'\in[k]$, there exists an edge between $G_c$ and $G_{c'}$. We show below that under these two conditions the optimum of UFM$_+$ under mini-batch SCL follows an OF geometry. In other words, we show that the optimal embeddings align if they belong to the same class (NC) and are orthogonal if they have different labels (mean-embeddings follow k-OF). (See Def. 3.)

*NC:* Consider a class $c$. From our assumption, the induced subgraph $G_c$ is connected. Thus, a path exists from any node (representing corresponding example) to any other node in $G_c$. Consider three nodes[6] along a path, indexed by $i,j,l$ that belong to class $c$. Let there be an edge between $i,j$ and $j,l$ each. By Def. 4 of the Batch Interaction Graph, examples $i$ and $j$ are present in a batch, say $B_1$, and examples $j,l$ belong to a batch $B_2$ that is possibly different from $B_1$. From Thm. 2, we can infer that at the optimal solution, we have $\mathbf{h}_i = \mathbf{h}_j$, due to equality conditions for batch $B_1$. Also, $\mathbf{h}_j = \mathbf{h}_l$, due to equality conditions for batch $B_2$. Thus, by transitivity, we have $\mathbf{h}_i = \mathbf{h}_j = \mathbf{h}_l$. Repeating this argument along any path in $G_c$ for every $c\in[k]$, we obtain the NC property: $\mathbf{h}_i = \mathbf{h}_j, \forall i,j : y_i = y_j = c$ for every class $c\in[k]$.

[6]The same arguments and considerations apply when $G_c$ consists of only two nodes.

*$k$-OF:* From our assumption, for any pair of classes $c_1$ and $c_2$, there exists an edge between $G_{c_1}$ and $G_{c_2}$ connecting at least one pair of nodes $i, j : i \in G_{c_1}, j \in G_{c_2}$. By definition of the Batch Interaction Graph, we know that examples $i$ and $j$ belong in at least one batch. Thus, by Thm. 2, at the optimal solution, we have $\mathbf{h}_i \perp \mathbf{h}_j$ since examples $y_i = c_1 \neq c_2 = y_j$. Now, by NC, $\mathbf{h}_i = \quad_{c_1}$, $\mathbf{h}_j = \quad_{c_2}$, and thus, $\quad_{c_1} \perp \quad_{c_2}$. This holds for every pair of classes $c_1 \neq c_2 \in [k]$. Therefore, the matrix $\mathbf{M} = [\quad_1, ..., \quad_k]$ of class-mean embeddings forms a $k$−OF.

Combining the two statements above, we conclude that at the global optimal solution, the embeddings follow the OF geometry, as desired.

• **'ONLY IF' direction.** For the other direction, it suffices to show that if either of the two conditions in Cor. 2.1 does not hold, then there exists an optimizer that does not follow the OF geometry. Specifically, we show that when either of the two conditions is violated and $d \geq k + 1$, there exists an embeddings matrix $\widetilde{\mathbf{H}}$ attaining the loss lower-bound that does not satisfy one of the two requirements of the OF geometry: 1) $\widetilde{\mathbf{H}}$ does not follow NC, or 2) the corresponding mean-embeddings $\widetilde{\mathbf{M}}$ do not arrange as a $k$-OF.

*Case 1.* Suppose for a $c \in [k]$, the induced subgraph $G_c$ is not connected, and without loss of generality, assume $c = 1$. This means that $G_1$ has at least two separate components. Denote the nodes in each of the two component by $V_1^1$ and $V_1^2$ respectively, and recall for $c \geq 2$, $V_c = \{i : y_i = c\}$. As $d \geq k + 1$, we can choose a set of $k + 1$ vectors $[\tilde{}_1^1, \tilde{}_1^2, \tilde{}_2, \tilde{}_3, ..., \tilde{}_k]$ such that they form a $(k + 1)$−OF. Define $\widetilde{\mathbf{H}} = [\widetilde{\mathbf{h}}_1, ..., \widetilde{\mathbf{h}}_n]$ as follows:

$$\forall i \in V_1^1, \quad \widetilde{\mathbf{h}}_i = \tilde{}_1^1$$
$$\forall i \in V_1^2, \quad \widetilde{\mathbf{h}}_i = \tilde{}_1^2$$
$$\forall i \in V_c, c \in [k], \quad \widetilde{\mathbf{h}}_i = \tilde{}_c .$$

Then, $\widetilde{\mathbf{H}}$ satisfies the equality conditions in Thm. 2 since there is no edge between the nodes in $V_1^1$ and $V_1^2$ by the assumption. However, the embeddings in class $y = 1$ do not align, since $\tilde{}_1^1$ is orthogonal to $\tilde{}_1^2$. Thus, $\widetilde{\mathbf{H}}$ optimizes UFM$_+$ while it does not satisfy NC and differs from the OF geometry.

*Case 2.* Suppose there exists $c_1 \neq c_2 \in [k]$ for which there is no edges between $G_{c_1}$ and $G_{c_2}$, and without loss of generality, assume $c_1 = 1, c_2 = 2$. Consider an embedding matrix $\widetilde{\mathbf{H}}$ that satisfies NC, and the corresponding mean-embedding matrix $\widetilde{\mathbf{M}} = [\tilde{}_1, \tilde{}_2, ..., \tilde{}_k]$ is such that $\forall c \neq c' \in \{2, ..., k\}$, $\tilde{}_c^\top \tilde{}_{c'} = 0$ and $\tilde{}_1 = \tilde{}_2$. Such an $\widetilde{\mathbf{M}}$ exists as $d \geq k$ and all we need is to have $[\tilde{}_2, ..., \tilde{}_k]$ be a $(k - 1)$−OF in the non-negative orthant. Since, there is no edge between $G_1$ and $G_2$, there is no $B \in \mathcal{B}$ that includes samples from classes $y = 1$ and $y = 2$ simultaneously. Equivalently, to achieve the lower-bound of Thm. 2, we do not require orthogonality between any pairs of samples $i \in \mathcal{C}_1$ and $j \in \mathcal{C}_2$. Therefore, $\widetilde{\mathbf{H}}$ is an optimal solution of the UFM$_+$. However, the mean-embeddings do not follow a $k$−OF. In fact, this optimal geometry, does not distinguish the samples in classes $y = 1$ and $y = 2$.

Therefore, the global minimizer will be uniquely an OF if and only if the batching scheme satisfies both the conditions stated in Cor. 2.1.

### B.2 Batch analysis for UFM$_+$ vs UFM

Graf et al. (2021, Thm. 2) studies the UFM without ReLU constraints and with balanced labels, and proves that the global solution is a simplex ETF. The authors note that their proof relies on the batch construction as the set of all combinations of a given size. In contrast, we have proved that for UFM$_+$ the global solution is an OF for a wider range of batching scenarios (specifically, as long as they satisfy the interaction properties characterized in Cor. 2.1). Without ReLU, as noted by Graf et al. (2021, Fig. 5), the optimal configuration of examples in each batch can have a different geometry, depending on the label distribution of the examples within the batch. However, *with* ReLU, the optimal configuration of every batch is an OF over the classes whose examples are present in the batch. In particular, there is no contradiction between two mini-batches having both overlapping and mutually exclusive classes, since the optimal configuration of one batch does not violate the optimal configuration of another. Furthermore, the overall batch construction can have a unique OF as the optimal configuration provided the conditions in Cor. 2.1 are satisfied. The conditions include the


**(a)** UFM optimal embeddings


**(b)** UFM$_+$ optimal embeddings

**Figure 6:** Rem. 1 visualized, **(a)(i,ii,iii)** indicate the antipodal structure of the optimal embeddings under UFM in each of the 3 mini-batches respectively, whereas the overall optimal geometry is an ETF **(a)(iv)**. This contrasts the optimal embeddings under UFM$_+$ where each mini-batch **(b)(i,ii,iii)** is consistent with the overall optima **(b)(iv)**.

batching assumed by Graf et al. (2021) as a special case, while being applicable in less restrictive scenarios. Sec. 5.2 explored the implications of this finding, by studying the criteria for a batching scheme to lead to a unique minimizer geometry and further suggesting a simple scheme to convert an arbitrary batching to one satisfying these criteria.

# C Proofs for Section D.1

## C.1 Proof of Lemma D.1

Let $\mathbf{V} = [\mathbf{v}_1 \quad \dots \quad \mathbf{v}_k]$ and $\mathbf{V} = [\mathbf{v}_1 \quad \dots \quad \mathbf{v}_k]$. By definition of a $k$-OF, we know that $\mathbf{V}^\top\mathbf{V} \propto \mathtt{I}$. Without loss of generality suppose the constant of proportionality is 1 so that $\mathbf{V}^\top\mathbf{V} = \mathtt{I}$. Since all $\mathbf{v}_i$s are orthonormal, this then implies that $\mathbf{V}^\top\mathbf{v}_G = \frac{1}{k}\mathbf{1}_k$ and $\mathbf{v}_G^\top\mathbf{v}_G = \frac{1}{k}$. These put together gives the desired as follows:

$$\mathbf{V}^\top\mathbf{V} = \left(\mathbf{V} - \mathbf{v}_G\mathbf{1}_k^\top\right)^\top\left(\mathbf{V} - \mathbf{v}_G\mathbf{1}_k^\top\right) = \mathbf{V}^\top\mathbf{V} - \mathbf{1}_k\mathbf{v}_G^\top\mathbf{V} - \mathbf{V}^\top\mathbf{v}_G\mathbf{1}_k^\top + (\mathbf{v}_G^\top\mathbf{v}_G)\mathbf{1}_k\mathbf{1}_k^\top = \mathtt{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^\top.$$

## C.2 Proof of Lemma D.2

To rule out the optimality of ETF in imbalanced setups, we show ETF does not minimize the loss in a $k = 3$ class example. Specifically, consider a STEP-imbalanced training set with 3 classes of sizes $[Rn_{\min}, n_{\min}, n_{\min}]$ with $n_{\min} \geq 2$ and $R \geq 10$. Suppose $\mathbf{H}_{\text{ETF}}$ follows the ETF geometry. Then, from Def. 5 and the feasibility condition, for the mean-embeddings $\mathbf{M}_{\text{ETF}}$ we have $\mathbf{M}_{\text{ETF}}^\top\mathbf{M}_{\text{ETF}} = \frac{k}{k-1}(\mathtt{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^\top)$. Thus, the value of the loss function at ETF is

$$\mathcal{L}_{\text{full}}(\mathbf{H}_{\text{ETF}}) = n_{\min}\Big(R\log\left(Rn_{\min} - 1 + 2n_{\min}e^{-\frac{3}{2}}\right) + 2\log\left(n_{\min} - 1 + n_{\min}(R+1)e^{-\frac{3}{2}}\right)\Big).$$

Now, consider $\widetilde{\mathbf{H}} = [\widetilde{\ } \otimes \mathbf{1}_{Rn_{\min}}^\top, -\widetilde{\ } \otimes \mathbf{1}_{n_{\min}}^\top, -\widetilde{\ } \otimes \mathbf{1}_{n_{\min}}^\top]$, where $\widetilde{\ }$ is an arbitrary unit-norm vector. Since $\|\widetilde{\ }\| = 1$, it follows that $\widetilde{\mathbf{H}}$ is in the feasible set. With this choice of $\widetilde{\mathbf{H}}$, we have,

$$\mathcal{L}_{\text{full}}(\widetilde{\mathbf{H}}) = n_{\min}\Big(R\log\left(Rn_{\min} - 1 + 2n_{\min}e^{-2}\right) + 2\log\left(2n_{\min} - 1 + Rn_{\min}e^{-2}\right)\Big).$$

It is easy to check that, for imbalance levels higher than $R \geq 10$,

$$\mathcal{L}_{\text{full}}(\widetilde{\mathbf{H}}) < \mathcal{L}_{\text{full}}(\mathbf{H}_{\text{ETF}}).$$

In other words, ETF does not always attain the optimal cost in UFM.

# D ADDITIONAL DISCUSSION

## D.1 DETAILED COMPARISON BETWEEN UFM AND UFM$_+$

Within this section, we provide an in-depth analysis of the disparities between the predictions generated by the UFM, commonly utilized in previous studies, and the refined version UFM$_+$ that we employ in this work to study the embedding geometry of SCL.

The majority of previous works (e.g., Papyan et al., 2020; Zhu et al., 2021; Fang et al., 2021) study the geometry of learned embeddings by investigating a proxy unconstrained features model (UFM). Specifically, the UFM drops the dependence of the learned embeddings $\mathbf{H}$ on the network parameters , and finds optimal $\widehat{\mathbf{H}}$ that minimizes the loss function. In case of SCL, as studied in e.g., Graf et al. (2021), the corresponding UFM is as follows,

$$\arg\min_{\mathbf{H}} \mathcal{L}_{\text{SCL}}(\mathbf{H}) \;\; \text{subj. to} \;\; \|\mathbf{h}_i\|^2 = 1, \; \forall i \in [n]. \tag{UFM}$$

As explained in Sec. 4, this paper employs a refined version of the UFM, namely UFM$_+$, to characterize the representations learned by SCL. Specifically, UFM$_+$ further constrains the embeddings $\mathbf{H}$ to be non-negative, in this way accounting for the ReLU activation commonly used in several deep-net architectures. Thus the search for the optimal $\mathbf{H}$ is restricted to the non-negative orthant.

### D.1.1 CENTERING HEURISTIC

Many deep neural network models incorporate ReLU activations, resulting in nonnegative feature embeddings at the last layer. In contrast, the UFM does not impose any constraints on the optimal $\mathbf{H}$, allowing it to contain negative entries. To bridge this gap, previous works (e.g., Papyan et al., 2020; Zhu et al., 2021; Fang et al., 2021; Zhou et al., 2022a; Thrampoulidis et al., 2022; Zhou et al., 2022b) apply a heuristic approach called *global-mean centering* on the learned embeddings before comparing their arrangement with the theoretical prediction given by the UFM. Specifically, the centering heuristic is applied as follows: Prior to comparing the geometries, we subtract the global-mean embedding $_G = \frac{1}{k}\sum_{c\in[k]} \; _c$ from all class-mean embeddings $_c$ to form the *centered class-mean embeddings* $_c$:

$$_c = \; _c - \; _G, \qquad \mathbf{M} = [\; _1 \; \cdots \; _k] \; .$$

Indeed, this heuristic centering operation effectively ensures that the mean-embedding matrix $\mathbf{M}$, after the necessary shifting, is centered at the origin, satisfying $\mathbf{M}\mathbf{1}_k = 0$. This property also holds for the global optimizers of the UFM found in the previous work, which served as a motivation for the heuristic centering.

Unlike the approach mentioned above, our findings do not necessitate any heuristic embedding processing for comparing the geometries. This is because our model directly provides the geometry of the embeddings in their original form.

### D.1.2 CENTERED OF IS SIMPLEX ETF

Focusing on the SCL, Graf et al. (2021) have used the UFM to characterize the learned embeddings and have found that, for balanced classes, the simplex ETF geometry is the global optimizer of UFM. In other words, the optimal embeddings $\mathbf{H}$ adhere to the NC (Defn. 1), and the corresponding mean embeddings $\mathbf{M}$ form a simplex ETF. For the reader's convenience, we recall below the definition of simplex ETF from Papyan et al. (2020).

**Definition 5** (Simplex ETF)**.** *We say that $k$ vectors $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_k] \in \mathbb{R}^{d\times k}$ form a simplex-ETF if $\mathbf{V}^\top\mathbf{V} \propto \mathrm{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^\top$, i.e., for each pair of $(i,j) \in [k]$, $\|\mathbf{v}_i\| = \|\mathbf{v}_j\|$ and $\frac{\mathbf{v}_i^T\mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|} = \frac{-1}{k-1}$.*

In this paper, we show instead that the global optimizer of UFM$_+$ is an OF. Remarkably, this result holds true regardless of the label-imbalance profile. However, for the purpose of comparison with the findings of Graf et al. (2021), let us specifically consider the scenario of balanced data. In this case, an intriguing question arises:

*Specifically for balanced data, how does our discovery that embeddings with ReLU converge to an OF compare to the previous finding by Graf et al. (2021) that embeddings without ReLU converge to an ETF?*

The answer to this question lies on the centering trick discussed in Section D.1.1. Specifically, stating that embeddings form an OF implies that *centered* embeddings form an ETF. This straightforward fact is formalized in the following lemma.

**Lemma D.1.** *Suppose that a set of vectors* $\{\mathbf{v}_1, ..., \mathbf{v}_k\}$ *form a* $k$*-OF in* $\mathbb{R}^d$ *with* $d \geq k$*, and their mean-centered counterparts are* $\mathbf{v}_c = \mathbf{v}_c - \mathbf{v}_G, \forall c \in [k]$*, where* $\mathbf{v}_G = \frac{1}{k}\sum_{c\in[k]} \mathbf{v}_c$*. Then, the centered embeddings* $\{\mathbf{v}_1, ..., \mathbf{v}_k\}$ *form a simplex ETF spanning a* $k-1$*-dimensional subspace.*

Lemma D.1 states that based on our finding that embeddings form an OF, we can deduce that centered embeddings form an ETF. This conclusion aligns with the analysis conducted by Graf et al. (2021) on the UFM (See Fig. 7 for a representative comparison). However, the UFM analysis itself does not provide information regarding the necessity of the centering technique, which remains heuristic. Additionally, it is important to note that it is unclear how to establish that (uncentered) embeddings form an OF if we initially know that centered embeddings form an ETF. This is due to the unknown vector used for centering, which cannot be determined.

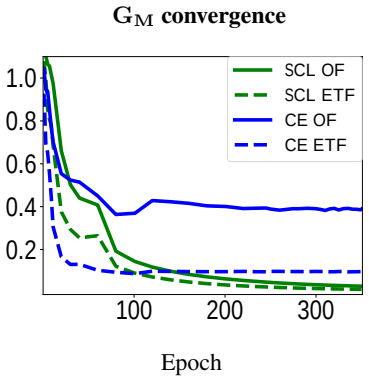



**Figure 7:** Convergence of embeddings geometries for SCL (in green) and CE (in blue) on balanced CIFAR10 with ResNet-18. Solid lines track the distance of *uncentered* embeddings to OF and dashed lines track the distance of *centered* embeddings to simplex ETF. For CE, the learning rate and weight decay and batch size are set according to Papyan et al. (2020). Note the convergence of SCL (solid green) to its implicit geometry is noticeably better compared to CE (dashed blue).

### D.1.3 UFM CAN FAIL TO PREDICT THE TRUE GEOMETRY

In the previous section, we demonstrated that for balanced data, the solution found by the UFM does not predict the true geometry of the embeddings (i.e., OF) in presence of ReLU, but it can predict the geometry of the embeddings after the application of a global-mean centering heuristic (i.e., ETF). Naturally this raises the following question:

*Does the UFM consistently provide accurate predictions for the geometry of the* centered *embeddings? In other words, does the OF geometry predicted by our refined* $\text{UFM}_+$ *always align with being a global optimizer of the UFM after centering?*

The lemma presented below demonstrates that the answer to this question is negative. Specifically, it shows that the global optimizer of UFM is sensitive to the label distribution of the training set, thus ETF is not necessarily a global optimizer in the presence of imbalances. This is in contrast to $\text{UFM}_+$, for which we showed that the global optimizer is consistently an OF, irrespective of the labels distribution. In other words, the difference between UFM and $\text{UFM}_+$ cannot be addressed by only considering the centering of the optimal embeddings in general.

**Lemma D.2.** *If classes in the training set are not balanced, the global solution of UFM is not necessarily an ETF. Thus, the solution of* $\text{UFM}_+$ *(which according to Thm. 1 is an OF) is in general different to that of UFM even after centering is applied.*

## D.2 COMPARING IMPLICIT GEOMETRIES: SCL VS CE – A SUMMARY

We conclude this discussion by providing a summary of the observations on the contrasting implicit geometries of embeddings between the two loss functions: SCL and CE.

**1. The SCL geometry with ReLU is robust to label-imbalances.** Specifically, the geometry of SCL embeddings exhibits symmetry consistently, whereas imbalances introduce distortions in the implicit geometry of SCL without ReLU and also for CE (Fang et al., 2021; Thrampoulidis et al., 2022).

**2. The convergence of SCL to its implicit geometry is notably superior.** The measured geometry of SCL exhibits a faster decrease in distance from its analytic formula (i.e., OF) with each epoch, eventually reaching lower values. This holds true not only for label-imbalanced data, where Thrampoulidis et al. (2022) finds that the convergence of embeddings deteriorates as the imbalance ratio increases, but also for balanced data, where the convergence of embeddings with CE is inferior

compared to classifiers (Papyan et al., 2020); see Fig. 7 for visualization.
**3. For SCL, batching matters.** During CE training, when using any arbitrary batching method, there exists an implicit interaction among all the examples, that is facilitated by the inclusion of the classifier vectors in the loss objective. On the other hand, with SCL, the interactions between examples are explicitly controlled by the mini-batches. Specifically, our analysis of the role of batches in Secs. 4.2 and 5.1, along with the batch-binding scheme presented in Sec. 5.2 reveal that the structure of mini-batches critically affects the geometry of the learnt embeddings.

# E ADDITIONAL EXPERIMENTAL RESULTS AND DISCUSSION

## E.1 DETAILS ON THE MAIN EXPERIMENTAL SETUP

In our deep-net experiments, we focus on two common network architectures, ResNet and DenseNet. Specifically, we use ResNet-18, ResNet-34 and DenseNet-40 with approximately 63, 11 million and 200 thousand trainable parameters, respectively. For all models, we replace the last layer linear classifier with a normalization layer (normalizing such that $\|\mathbf{h}_i\| = 1$ for $i \in [n]$) of feature dimension $d = 512$. Specifically, following Graf et al. (2021), we directly optimize the normalized features that are then used for inference. (We note this is slightly different compared to Khosla et al. (2020), where the authors train the output features of a projection head, then discard this projection head at inference time). We use this projection head when studying NCC test accuracy. In particular, we add a 2 layer non-linear MLP with with and without ReLU to report the test accuracies provided in Tab. 1, Tab. 2 and , Tab. 5. Following Khosla et al. (2020), in addition to the NCC test accuracy on post-projection feature (final output), we evaluate the accuracies on the pre-projection features (without input to projector) as well. We remark that both ResNet and DenseNet architectures include ReLU activations before the final output, which enforces a non-negativity on the learned embeddings. Resnet-18 models with CIFAR10, MNIST have been trained for 350 epochs with a constant learning rate of 0.1, no weight decay, batch size of 1024, and SCL temperature parameter $\tau = 0.1$ (consistent with the choice of $\tau$ made in Khosla et al. (2020); Graf et al. (2021)). For CIFAR100 and Tiny-ImageNet experiments, we train a ResNet-34 under a similar setup, for 500 epochs, (with batch binding in Fig. 2). For all experiments with other models or datasets, we provide experimental details in the following sections. All ResNet-18 and DenseNet models have been trained on a single Tesla V100 GPU machine and all ResNet-34 models were trained on a single machine with 2-4 Tesla V100 GPU, depending on experiment.

## E.2 DETAILS ON FIG. 2 HEATMAPS

Since this particular experiment provides the most insight into the geometric behaviour of features, we have provided further explanation of the experimental setup and method. ResNet-18 models were trained for 350 epochs on MNIST using **CE**, **SCL** and **SCL + ReLU** under difference imbalance ratios. For SCL and SCL + ReLU we extract the features, calculate the class means and plot a heatmap of the gram matrices $G_M$. For CE, following previous work by Papyan (2018); Thrampoulidis et al. (2022); Zhu et al. (2021) we center the class means by the global mean, ie $\overline{\mu_c} = \mu_c - \frac{1}{K}\sum_{i=1}^{K}\mu_i$ and then proceed to calculate $G_M$. In order to have a fair comparison, we normalize all gram matrices by dividing by the matrix maximum $\overline{G_M} = \frac{G_M}{\max_{i,j}|G_\mu^{i,j}|}$. No projection layer was implemented for these results. The heatmaps were included to provide a clear visual, emphasizing the impact of ReLU on achieving symmetric geometry under imbalance. Each cell $[i, j]$ of the $G_M$ matrix represents the inner product $\mu_i^T \mu_j$, rather a normalized value representing the angle between the two class centers. In addition, as SCL includes a normalization layer, the diagonals $G_M[i, i]$ would be closer to zero as the features exhibit stronger neural collapse properties.

## E.3 ADDITIONAL GEOMETRIC ANALYSIS

In addition to analyzing the convergence of the Gram-matrix of class-mean embeddings $\mathbf{G_M}$ to the OF geometry (as provided in Fig. 2), we also keep track of Neural Collapse (Defn. 1) of individual embeddings and orthogonality of their class-means (Defn. 2) separately. Furthermore, we qualitatively study the Gram matrices $\mathbf{G_M}$ and $\mathbf{G_H}$ and compare them to corresponding matrices for the CE loss under imbalances.

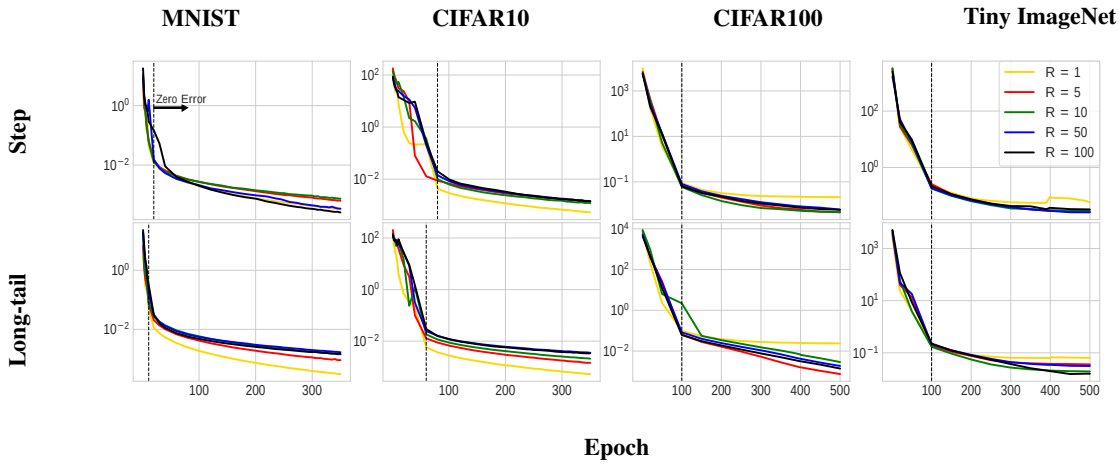



**Figure 8:** Neural Collapse metric $_{\text{NC}} := \mathsf{tr}(\ _W\ ^{\dagger}_{B})/k$ for the corresponding experiments in Fig. 2. Values are usually on the order of $10^{-2}$ suggesting strong convergence of embeddings to their class means.

### E.3.1 Neural Collapse

In order to quantify the collapse (NC) of embeddings, $\mathbf{h}_i, i \in [n]$, to their class-means, we measure $\beta_{\text{NC}} := \mathsf{tr}(\ _W\ ^{\dagger}_{B})/k$ (Papyan et al., 2020). Here, $\ _B = \sum_{c\in[k]}(\ _c - \ _G)(\ _c - \ _G)^{\top}$ is the between class covariance matrix, $\ _G = \frac{1}{k}\sum_{c\in[k]}\ _c$ is the global mean, and $\ _W = \sum_{i\in[n]}(\mathbf{h}_i - \ _{y_i})(\mathbf{h}_i - \ _{y_i})^{\top}$ is the within class covariance matrix.

For the experiments shown in Fig. 2 and Fig. 3 in the main body, the corresponding values of $\beta_{\text{NC}}$ can be found in Fig. 8 and Fig. 12(b) respectively.

### E.3.2 Angular convergence

Having confirmed convergence of embeddings to their respective class-means via NC ($\mathbf{h}_i \approx \ _c$ for $i : y_i = c$), we can now compare the feature geometry to the OF geometry by calculating the average $\alpha_{\text{sim}}(c, c') := \ ^{\top}_{c}\ _{c'}/\|\ _c\|\|\ _{c'}\|$ between each pair of classes. Fig. 9 plots the average cosine similarity $\text{Ave}_{c\neq c'}\alpha_{\text{sim}}(c, c')$ between class means for the same experiment as that of Fig. 2. The graphs indicate strong convergence to orthogonality between feature representations of different classes. Similarly, results for angular convergence corresponding to Fig. 3 are provided in Fig. 12(c), indicating similar convergence to OF for the full-batch experiments.

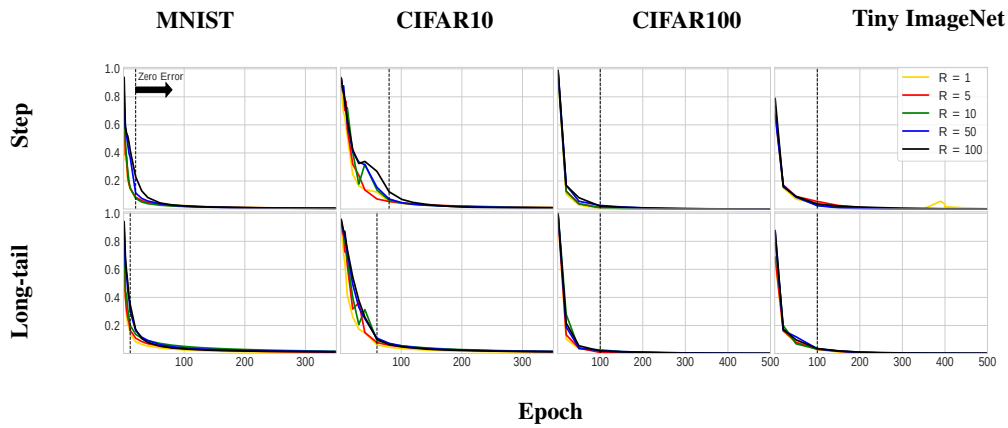



**Figure 9:** Average cosine similarity between different class means $_{\text{sim}}(c; c') := \boldsymbol{\mu}_c^{\top}\boldsymbol{\mu}_{c'}/\|\boldsymbol{\mu}_c\|\|\boldsymbol{\mu}_{c'}\|$ for corresponding results from Fig. 2. Final values are mostly on the order of $10^{-2}$, indicating strong orthogonality between class-mean embeddings.

### E.3.3 EMBEDDING HEATMAPS

As a qualitative measure, we have generated heatmaps that visually represent the learned embedding geometries; see Figs. 1, 10, 11. Specifically, we generate heatmaps of the Gram-matrices $\mathbf{G_M} = \mathbf{M}^\top\mathbf{M}$ and $\mathbf{G_H} = \mathbf{H}^\top\mathbf{H}$. In Fig. 1 we train ResNet-18 with the full MNIST dataset. In Fig. 10 we run on a subset of the dataset ($n = 10000$ total examples) with a batch size of 1000. Specifically for Fig. 10, when optimizing with CE loss we modify the network (ResNet-18) such that it has a normalization layer before the linear classifier head. We consider this additional setting to allow for a comparison where CE features are constrained to the unit sphere akin to our SCL experiments. Lastly, in Fig. 11 we plot the learned features Gram-matrix $\mathbf{G_H}$ for a ResNet-18 trained on CIFAR10 ($n = 10000$ total examples) with a batch size of 1000.[7] This heatmap qualitatively shows a more complete picture as we are plotting $\mathbf{G_H} = \mathbf{H}^\top\mathbf{H}$ rather than $\mathbf{G_M}$, thus simultaneously illustrating both Neural Collapse and convergence to the k-OF structure.

### E.3.4 EXPERIMENTS WITH MLPS

In Fig. 14 we run experiments with a simple 6 layer multilayer perceptron (MLP) to further explore the impact of model complexity on geometric convergence. The MLP includes batch normalization and ReLU activation between each layer. Each layer has 512 hidden units. We train the model with a batch size of 1000 with random reshuffling at each epoch. Furthermore, we train under $R$-STEP imbalanced MNIST. No batch duplication was used. All other aspects of the implementation are as described in Sec. E.1. As shown in Fig. 14 all metrics $_{\mathbf{G_M}}$, $\beta_{\text{NC}}$, and $\text{Ave}_{c\neq c'}\ \alpha_{\text{sim}(c;c')}$ indicate strong convergence to the OF geometry, irrespective of imbalance ratio $R$.

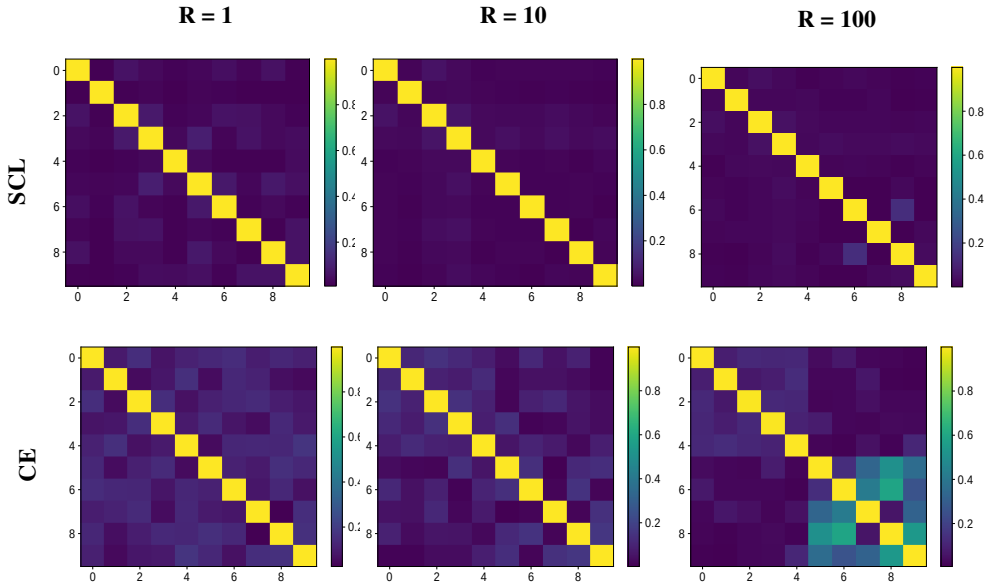



**Figure 10:** $\mathbf{G_M}$ Gram-matrices of mean-embeddings for various $R$-STEP imbalances at last epoch (350) of training with ResNet-18 on MNIST with $n = 10000$. To allow for fair comparison, the CE features are normalized before the classifier head akin to the SCL experiments.

## E.4 OPTIMIZATION DYNAMICS

### E.4.1 LOSS CONVERGENCE

In order to compute the lower bounds shown in Fig. 3, we use Thm. 1, substituting $e^{-1}$ with $e^{-1/}$ using $\tau = 0.1$ as employed in our experiments; this substitution is allowed thanks to Remark 2. Furthermore, we compute the lower bounds and $\mathcal{L}(\mathbf{H})$ on a per sample basis, thus we divide by $n = 1000$ which corresponds to the number of datapoints in our single batch. Lastly, as a method of

[7] In both Fig. 11 and Fig. 10 no batch duplication is used as described in Sec. 3.

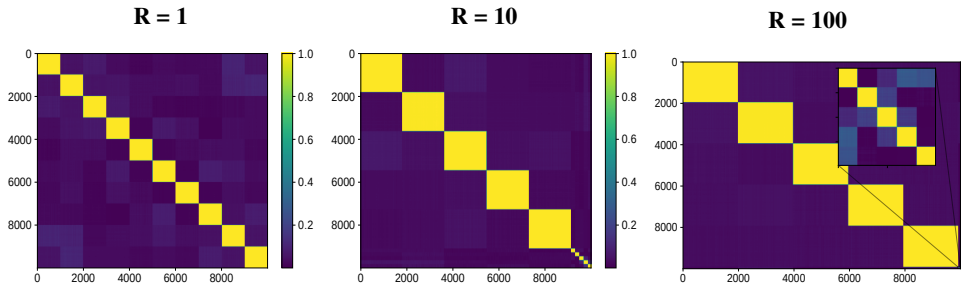



**Figure 11:** $\mathbf{G_H}$ Gram-matrices of feature embeddings for various imbalances at last epoch (350) of training SCL with CIFAR10 and ResNet.

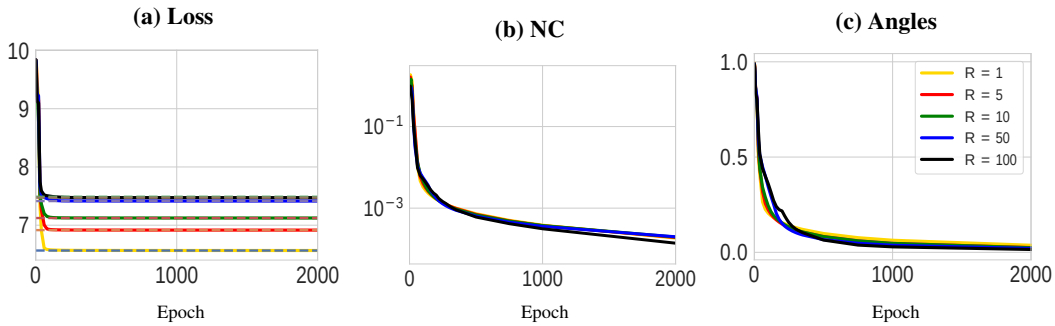



**Figure 12:** Full-batch SCL: ResNet-18 trained on a $R$-STEP imbalanced subset of MNIST of size $n = 1000$. **(a)** Loss converges to the lower bound (dashed lines) computed in Thm. 1. **(b)** Within-class feature variation becomes negligible (NC). **(c)** The average pairwise cosine similarity of class-means approaches zero. Each epoch is equivalent to one iteration of gradient descent.

maintaining numerical stability (as implemented in Khosla et al. (2020)) we apply a global scaling of the loss by a factor $/_{b}$, where $\tau_b = 0.07$ is the base temperature Khosla et al. (2020). The complete experiment, conducted over 2000 epochs (with axes limited to 500 epochs for clarity in Figure 3), is available in Figure 12.

### E.4.2 Effect of $\tau$

**Remark 2.** *The normalization of the embeddings in UFM$_+$ corresponds to SCL with temperature $\tau = 1$. More generally, the normalization becomes $\|\mathbf{h}_i\|^2 = 1/\tau$. The conclusion of Thm. 1 is insensitive to the choice of $\tau$, thus stated above for $\tau = 1$ without loss of generality. Although the value of $\tau$ does not affect the global optimizers of UFM$_+$, we have empirically observed that it impacts the speed of convergence during training.*

As described in Sec. 3 and Sec. 4.1 the optimality of the OF geometry for the UFM$_+$ is invariant to the choice of the temperature parameter $\tau$. However, we have found that the speed of convergence to the OF geometry is dependent on the choice of $\tau$. Shown in Fig. 13 is a full batch SCL experiment on $n = 1000$ samples of MNIST trained on ResNet-18 with $\tau = 0.1, 1, 10$. It is clear from Fig. 13 that: (a) the within-class feature variation converges significantly faster for smaller $\tau$; (b) the angles converge to k-OF faster and smoother for smaller $\tau$ as well (see Fig. 13 (b)). In all cases, values of $\beta_{\text{NC}}$ and $\alpha_{\text{sim}}(c, c')$ continue to decrease, and we anticipate further convergence if the networks are trained for longer. These results qualitatively agree with the findings of Khosla et al. (2020) which suggest that smaller $\tau$ improves training speed.

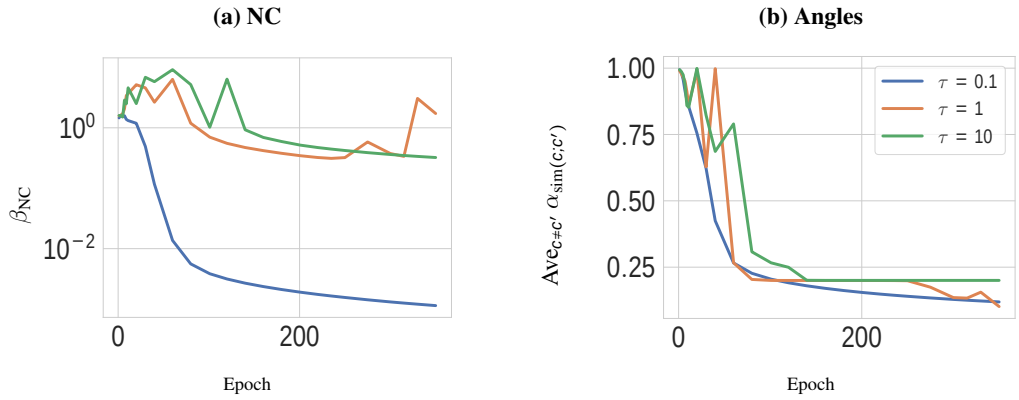



**Figure 13:** Full-batch SCL: ResNet-18 trained on a subset of MNIST of size $n = 1000$ with different temperature parameters . **(a)** Within-class feature variation (NC). **(b)** Average pairwise cosine similarity of class-means. Each epoch is equivalent to one iteration of gradient descent.

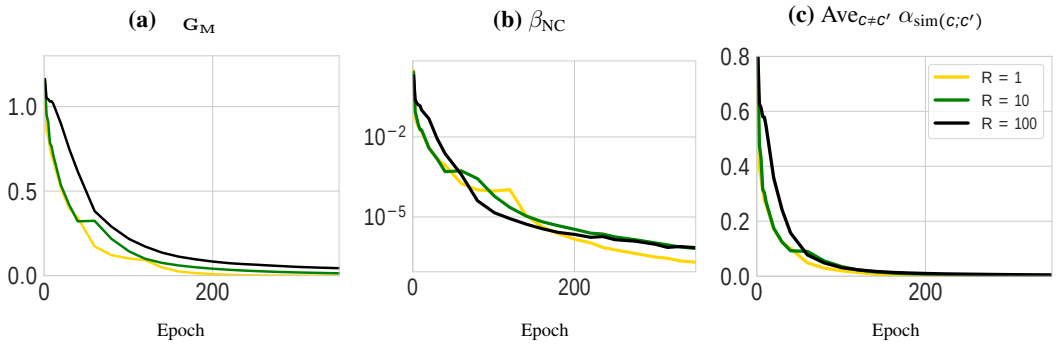



**Figure 14:** Geometry convergence metrics **(a)** $\mathbf{G}_M$, **(b)** NC, and **(c)** $\text{Ave}_{c\neq c'}$ sim$(c;c')$ for a 6 layer multilayer perceptron (MLP) model with ReLU activations trained with SCL and MNIST under $R$-STEP imbalance.

### E.4.3 Impact of Batch Size

Khosla et al. (2020) have found performance increases with larger batch sizes, and thus we consider the effects of batch size on the convergence of the learned features to their corresponding predicted geometry. We see in Fig. 15 that the features learned when ReLU is applied are more robust to variations in the batch size. Without ReLU, despite having ETF as the optimal geometry, as predicted by Graf et al. (2021), convergence is considerably more sensitive to batch size choice. This sensitivity is exacerbated when the number of classes increases, see CIFAR100 in Fig. 15.

## E.5 Complementary results and discussions on batch-binding

In this section, we describe examples where batching methods and batch-binding help improve the convergence speed of embeddings geometries to OF.

### E.5.1 How batch-binding ensures a unique OF geometry

Fig. 16 provides a simple illustration demonstrating how adding binding examples can satisfy the requirements of Cor. 2.1. While there are alternative approaches to satisfy the graph conditions stated in Cor. 2.1 for ensuring a unique OF geometry, the method of adding the same $k$ examples to each batch is a straightforward technique that is often computationally efficient, considering that batch sizes typically exceed the number of classes $k$.

**Geometry convergence for different batching schemes.** A comparison of OF convergence with three batch selection schemes was introduced in Sec. 5. We considered three strategies: 1) **No batch-shuffling**, 2) **Batch-shuffling**, and 3) **No batch-shuffling + batch-binding**. We noted that

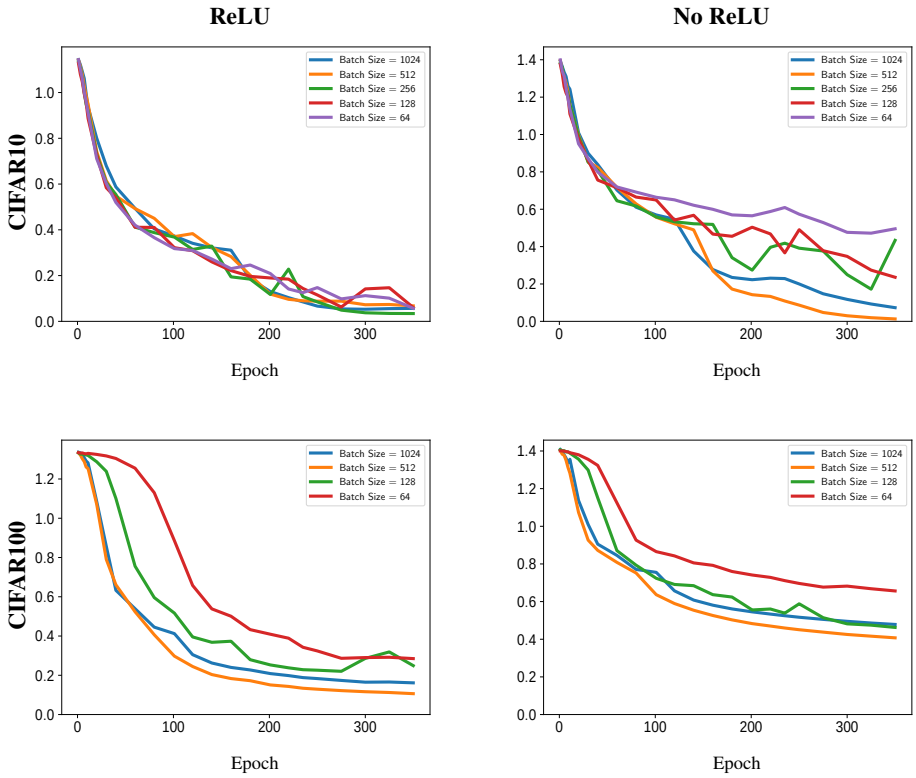



**Figure 15:** Convergence of learned features Gram matrices, $\mathbf{G}_{\mathbf{M}}$ to predicted geometry; OF in when ReLU is applied to features and ETF otherwise. Batch size is varied with balanced data.

optimization with batch-binding, even in absence of shuffling is very effective at guiding the solution to the unique OF geometry.

Further, in Table 5 in the appendix, we present experimental evidence in suggesting that batch binding does not negatively effect the balanced test accuracy of *Nearest Class Center* (NCC) classifier. This observation, paired with the impact of binding examples on OF convergence, motivates further analysis of the relationship between OF geometry and test accuracy.

**Improving convergence for less powerful models.** When conducting our experiments, we noticed that DenseNet-40 converges slower compared to ResNet-18. A reason for this may be related to the very different complexities of the two models: ResNet-18 has substantially more trainable parameters compared to DenseNet-40. In an attempt to improve the convergence speed, we reduce the batch size for DenseNet experiments to 128 to increase the number of SGD iterations. We also train DenseNet for 500 epochs instead of 350 while reducing the learning rate from 0.1 to 0.01 at epoch 300. Yet, we observe that with much smaller batch sizes, the embedding geometry does not always converge to the OF geometry, especially when training with high imbalance ratios. Specifically when training with randomly reshuffled batches, there is a higher chance that the examples do not interact with each other even if trained for long, suggesting that the optimal solution for all batches is not necessarily OF. We hypothesize that this likelihood increases when using smaller batch sizes during training. In order to combat this effect, we ran the DenseNet experiments with the addition of binding examples to every batch. As shown in Fig. 19, we find that adding such binding examples significantly improves the convergence to OF, particularly when training on more complex datasets (CIFAR10 compared to MNIST) and under more severe imbalances (STEP imbalance with $R = 50, 100$). These results emphasize the impact of batch-binding and provide further evidence regarding our claims in Sec. 4.2. While the convergence values of $\mathbf{G}_{\mathbf{M}}$ are higher when compared to ResNet-18, we deem them reasonable considering the difference in the number of parameters between the two architectures. Similar to ResNet-18, we provide geometric convergence results for DenseNet-40 on different datasets in Fig. 23 in the appendix.

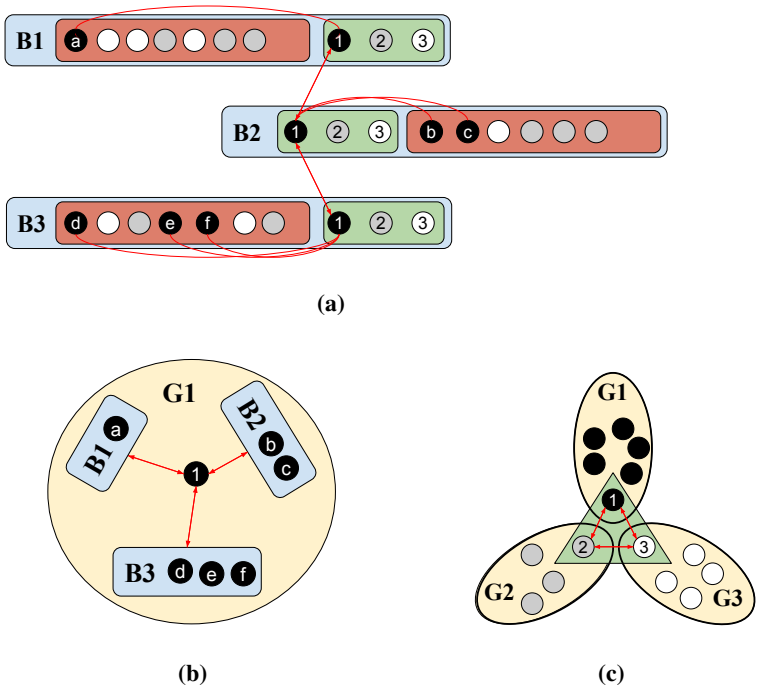



**Figure 16:** A simple illustration to explain how adding binding examples to each batch satisfies the requirements of Cor. 2.1, thus leads to unique OF geometry. **(a)** Gives a 3 class (black, grey and white) classification example with 3 batches. In addition to the data for each batch (included in red), the binding examples $1; 2; 3$ are added to each batch. **(b)** Gives the Batch Interaction Graph for the induced subgraph $G_1$ of the class (black) of example 1 and illustrates how all batches are connected through examples 1, satisfying the first condition of Cor. 2.1. **(c)** Elaborates on how all three class graphs $G_1; G_2; G_3$ are connected through the interactions of data points $1; 2; 3$, satisfying the second condition of Cor. 2.1.

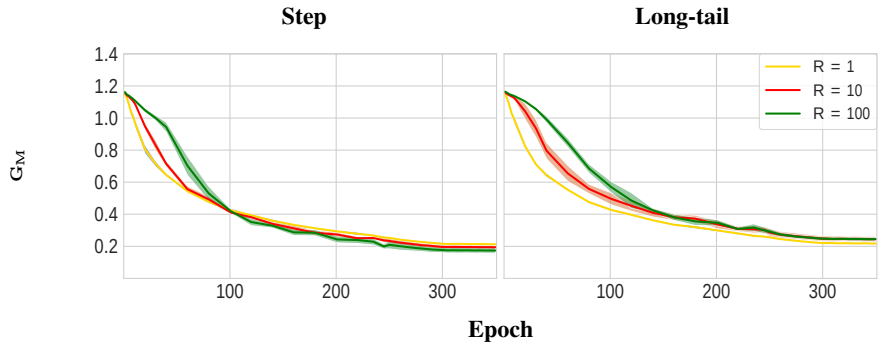



**Figure 17:** Convergence of $\mathbf{G}_{\mathbf{M}}$ to the OF geometry for a ResNet-18 model trained on CIFAR10 under Step and Long-tail imbalance, with data augmentation. Results represent the average run results over 5 versions of the experiments with randomly chosen binding examples from each class.

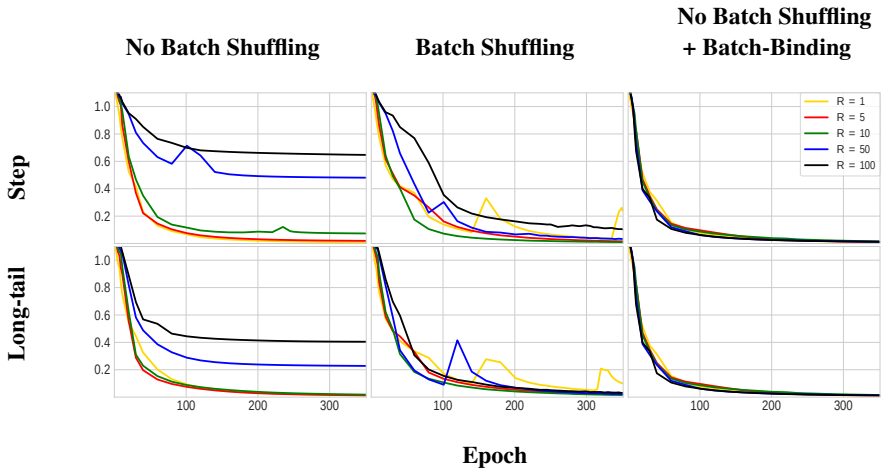



**Figure 18:** Convergence to the OF geometry for various batching schemes including the analysis-inspired scheme "No Batch Shuffling + Batch-Binding". See text for details. Experiments conducted with CIFAR-10 and ResNet-18.

In another experiment, we consider the convergence of ResNet-18 embeddings to the OF geometry when trained with CIFAR100. Models are trained for 500 epochs with a constant learning rate of 0.1 and a batch size of 1024. In Fig. 2, we see that ResNet-18 easily converges to CIFAR10. However, without batch binding, in Fig. 20 ResNet-18 struggles to convergence. With a large number of classes ($k$ = 100 for CIFAR100), it becomes increasingly more likely for the randomly reshuffled batches to not allow sufficient interactions between examples. Particularly for large imbalance ratios (e.g., $R$ = 100, 50), since the number of samples in each minority class could become as low as 5-10, some batches might never encounter examples from minority classes. This is particularly noticeable for STEP-Imbalance and large imbalance ratios, where half of the classes are minorities. On the other hand, including the binding examples in each batch (right side) can improve the convergence of the feature geometry to OF.

### E.5.2 Convergence in non-ReLU settings

In addition to observing that batch-binding improves the convergence of SCL with ReLU to OF, we consider the setting without ReLU. In particular in Fig. 21 we compare the convergence of the Gram matrix of mean embeddings to the ETF geometry with and without batch-binding. We train a ResNet-18 model with CIFAR-10. We observe that with a reduced batch size of 128 the convergence can be significantly improved with binding-examples.

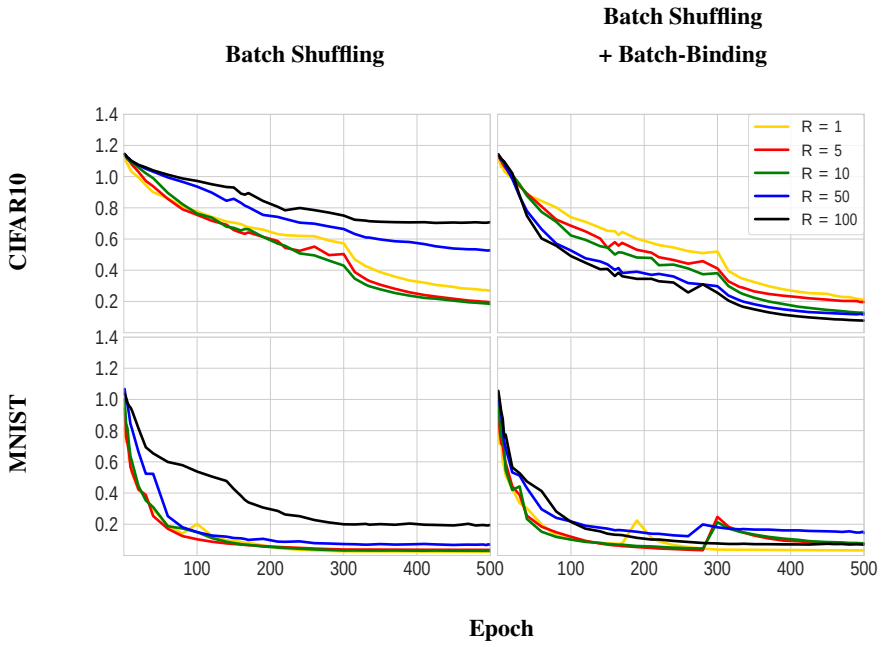



**Figure 19:** Convergence of $\mathbf{G}_{\mathbf{M}}$ to the OF geometry for a DenseNet-40 model trained on CIFAR10 and MNIST under STEP imbalance with and without batch-binding. While the impact of batch-bindings is less noticeable when training on a simple dataset such as MNIST, the convergence is significantly improved, particularly under imbalance, when training on a more complex dataset such as CIFAR10.

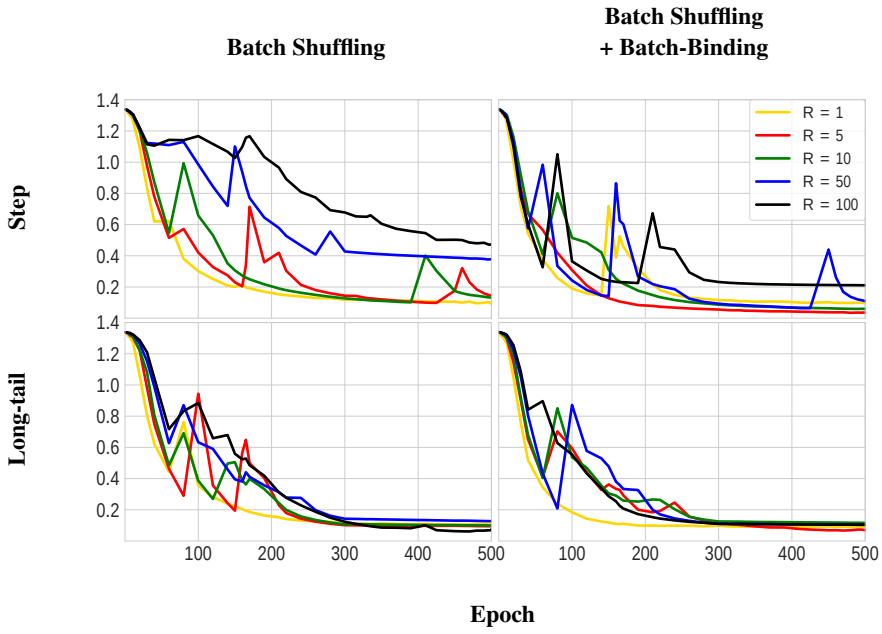



**Figure 20:** Convergence of embeddings geometry to OF as measured by $\mathrm{G}_{\mathbf{M}} := \| \frac{\mathbf{G}_{\mathbf{M}}}{\|\mathbf{G}_{\mathbf{M}}\|_F} - \frac{\mathrm{I}_k}{\|\mathrm{I}_k\|_F} \|_F$ for ResNet-18 trained on imbalanced CIFAR100 with SCL and different batching schemes. Adding binding examples helps with the convergence to the OF geometry, especially under STEP imbalance with larger imbalance ratios.

### E.6 On the convergence of batch-shuffling to OF

It can be readily observed that a fixed mini-batch partition without shuffling (*No batch-shuffling* in the experiments above) does not satisfy the conditions of Cor. 2.1. Consequently, OF is *not* the unique

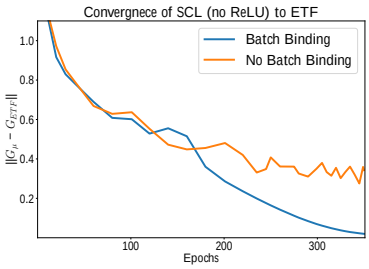



**Figure 21:** Geometry convergence of $\mathbf{G} = \mathbf{M}^\top \mathbf{M}$ to the ETF geometry with and without binding examples. A batch size of 128 is used, there is no ReLU on the output features.

minimizer geometry in such scenarios. This is clearly manifested by the inadequate convergence levels observed in our experiments, as depicted for example in Fig. 18. In contrast, in our experiments, when the examples are randomly shuffled prior to partitioning into mini-batches at each epoch (which we call *Batch Shuffling*), the convergence behavior to OF shows a significant improvement compared to the case of *No batch-shuffling*. The observed improvement can be attributed to the fact that shuffling enables interactions among examples across epochs. To elaborate on this notion, we present a formalization of batch shuffling below.

Let $\mathcal{S}$ denote the set of all permutations of $[n] = \{1, 2, ..., n\}$, where $n$ is the total number of training examples. Additionally, let $b$ denote a fixed batch-size. For simplicity, assume $n$ is an integer multiple of $b$. At the beginning of each training epoch $t$, we sample a permutation $s_t = (i_1, i_2, ...., i_n)$ uniformly from $\mathcal{S}$ *with* replacement. We then define $\mathcal{P}(s_t) := \{\{i_1, ..., i_b\}, \{i_{b+1}, ..., i_{2b}\}, ..., \{i_{n-b+1}, ..., i_n\}\}$ as the collection of total $\frac{n}{b}$ mini-batches obtained by partitioning $s_t$. These are the mini-batches used at epoch $t$. Thus, SCL in this epoch can be written as follows:

$$\mathcal{L}_{s_t}(\mathbf{H}) := \sum_{B \in \mathcal{P}(s_t)} \sum_{i \in B} \mathcal{L}_B(\mathbf{H}; i), \quad \text{where } \mathcal{L}_B(\mathbf{H}; i) := \frac{1}{n_{B;y_i} - 1} \sum_{\substack{j \in B \\ j \neq i; y_j = y_i}} \log\Big( \sum_{\substack{\ell \in B \\ \ell \neq i}} \exp\big(\mathbf{h}_i^\top \mathbf{h}_\ell - \mathbf{h}_i^\top \mathbf{h}_j\big)\Big),$$

and recall $n_{B;c} = |\{i : i \in B, y_i = c\}|$. Consider also the loss over *all* mini-batches obtained by partitioning each permutation of $\mathcal{S}$:

$$\mathcal{L}_{\mathcal{S}}(\mathbf{H}) := \sum_{s \in \mathcal{S}} \sum_{B \in \mathcal{P}(s)} \sum_{i \in B} \mathcal{L}_B(\mathbf{H}; i), \tag{10}$$

Now, since $s_t$ is uniformly sampled from $S$, we have the following relation for the expected per-epoch loss to the total loss given in Eqn. (10): $\mathbb{E}_{s_t} \mathcal{L}_{s_t}(\mathbf{H}) = 1/|\mathcal{S}| \, \mathcal{L}_{\mathcal{S}}(\mathbf{H})$. Therefore, training with batch-shuffling can be regarded as a stochastic version of minimizing the total loss in Eq. (10). Regarding the latter, it can be checked (see Rem. 3 below) that it satisfies the conditions of Cor. 2.1, thereby making OF its unique minimizer geometry. Taken together, these findings suggest that although the per-epoch loss $\mathcal{L}_{s_t}$ obtained through batch-shuffling does not satisfy the conditions of Cor. 2.1 for any specific epoch $t$, it does satisfy them in expectation. In particular, this can be regarded as preliminary justification of the experimental observation of improved convergence *with batch-shuffling* compared to *No batch-shuffling* schemes. The latter schemes fail to satisfy Cor. 2.1 under any circumstances, unless in the extreme case of a large batch size where $b = n$ and we recover $\mathcal{L}_{\text{full}}$. However, it is important to note that the aforementioned argument is a rough approximation, as it is not feasible to average over all $|\mathcal{S}| = n!$ permutations when optimization is typically performed over a limited number of epochs. This can explain why, despite exhibiting better convergence compared to the No batch-shuffling schemes, batch-shuffling schemes still demonstrate non-smooth and inconsistent convergence patterns in our experiments. This behavior becomes particularly evident when comparing the convergence levels of batch-shuffling to our *batch-binding* scheme, which is specifically designed to satisfy the conditions of Cor. 2.1 at *each* epoch.

**Remark 3.** *To show that Eqn.* (10) *satisfies Cor. 2.1, we will show that the corresponding* batch interaction graph *$G$ is a complete graph. This suffices, because the induced subgraphs $G_c$ from Def. 4 for a complete graph are connected graphs, and there exist multiple edges between $G_{c_1}$ and $G_{c_2}$ for $c_1, c_2 \in [k]$. To show that $G$ is a complete graph, we argue as follows. Consider the set $\mathcal{B}$ of all mini-batches obtained by partitioning all $n!$ elements of $\mathcal{S}$. Fix $b \geq 2$, so that the mini-batches have at least*

*two examples. We will consider a subset $\mathcal{B}_1 \subseteq \mathcal{B}$ of mini-batches and show that the corresponding batch interaction graph for $\mathcal{B}_1$ is a complete graph, which would then imply the batch interaction graph of $\mathcal{B}$ is also a complete graph. Specifically, let $\mathcal{B}_1 \subseteq \mathcal{B}$ denote the mini-batches comprising of the first $b$ indices in every permutation $s \in \mathcal{S}$. In other words, from a given $s = (i_1, i_2, ..., i_n) \in \mathcal{S}$, we let $\mathcal{B}_1$ include the mini-batch $\{i_1, i_2, ..., i_b\}$. Since $\mathcal{S}$ includes all permutations of $[n]$, $\mathcal{B}_1$ contains all $b$-length permutations of the elements of $[n]$, possibly with repetitions. Thus, the* batch interaction graph *created by the mini-batches in $\mathcal{B}_1$ is a* complete graph. *In the definition of the* batch interaction graph*, since a repeated presence of a pair of examples does not alter the graph, the batch interaction graph for $\mathcal{B}$ is the same complete graph. Thus, $\mathcal{B}$ satisfies Cor. 2.1.*

### E.7 ReLU does not compromise Accuracy

Having discussed the symmetry-preserving value of ReLU activation in SCL training, we now turn towards the impact on generalization accuracy. While the exact relationship between the training feature geometry and generalization is an open question, it is conceivable that maintaining symmetry in features is beneficial to generalization (Behnia et al., 2023; Zhu et al., 2022). In fact, our experiments with class-imbalance reveal that ReLU improves the test accuracy significantly in a subset of the cases, while not deteriorating in others. There have been suggestions of loss function modifications in such a way that the feature geometry is symmetric in presence of class-imbalance, consequently also achieving improvements in generalization Behnia et al. (2023); Zhu et al. (2022). Since the role of a projector block is of interest when training with SCL, we compare the experimental test accuracies both when the projector is retained (post-projection) as well as discarded (pre-projection) during inference (Khosla et al., 2020; Chen et al., 2020; Zhu et al., 2022).

| | | Pre-Projection | | | | Post-Projection | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | MLP | | MLP + ReLU | | MLP | | MLP + ReLU | |
| | R | Step | LongTail | Step | LongTail | Step | LongTail | Step | LongTail |
| CIFAR-10 | 1 | 91.88 ± 0.29 | | 92.04 ± 0.10 | | 91.94 ± 0.04 | | 91.79 ± 0.13 | |
| | 10 | 83.70 ± 1.09 | 83.82 ± 0.70 | 83.41 ± 0.64 | 84.40 ± 0.79 | 82.35 ± 0.67 | 82.97 ± 0.83 | 80.81 ± 1.01 | 82.99 ± 1.03 |
| | 100 | 60.19 ± 1.75 | 68.75 ± 0.31 | 67.12 ± 1.04 | 68.57 ± 1.69 | 55.31 ± 1.98 | 63.67 ± 0.77 | 59.83 ± 1.75 | 62.65 ± 2.37 |
| CIFAR-100 | 1 | 72.17 ± 0.23 | | 72.32 ± 0.60 | | 72.30 ± 0.57 | | 72.25 ± 0.35 | |
| | 10 | 56.58 ± 0.50 | 57.10 ± 1.04 | 58.16 ± 0.76 | 57.88 ± 1.05 | 54.57 ± 0.49 | 56.27 ± 0.45 | 55.35 ± 0.33 | 57.08 ± 0.82 |
| | 100 | 43.49± 0.30 | 37.19 ± 2.50 | 43.80 ± 0.25 | 39.71 ± 0.09 | 40.29 ± 0.33 | 35.29 ± 1.64 | 39.93 ± 0.33 | 37.21 ± 0.51 |
| Tiny ImageNet | 1 | 62.53 ± 1.23 | | 62.26 ± 0.37 | | 62.71 ± 1.10 | | 62.48 ± 0.40 | |
| | 10 | 50.17 ± 1.44 | 50.8 ± 1.94 | 49.78 ± 0.67 | 49.81 ± 0.82 | 48.91 ± 1.6 | 50.23 ± 1.83 | 48.23 ± 0.53 | 48.76 ± 1.18 |
| | 100 | 39.13 ± 0.63 | 36.06 ± 0.89 | 40.02 ± 0.62 | 35.84 ± 1.33 | 37.41 ± 0.29 | 34.57 ± 0.95 | 37.19 ± 0.68 | 33.78 ± 1.34 |

**Table 2:** Comparison of test accuracies for (i) CIFAR-10 (ii) CIFAR-100 (iii) Tiny ImageNet when ResNet-18 is trained with a 2-layer MLP projection head with and without ReLU on the output features. Comparisons in both Step and LT imbalance settings. Additionally a comparison of the performance using features before the projection head (pre-projection), and after the head (post-projection).

Thus, as noted in Table 2 we make a remarkable observation that simply adding ReLU to the network can yield significant improvements in test accuracy in certain cases, while at least matching that of the case without ReLU in others. In light of this, our finding of the insensitivity of the embedding geometry in presence of ReLU serves a potential explanation. Our analysis therefore paves an important direction for future empirical investigations into the role of the activation functions in the projector network on generalization of SCL.

### E.8 ReLU Helps improve worst class accuracy

As illustrated by Fig. 2 and Fig. 1, the use of ReLU results in a symmetric OF feature geometry and prevents the collapse of the minority feature vectors. While the direct relationship between NC geometry of train and test features is not yet entirely established, we expect that this symmetric

geometry could help improve the worst class test accuracies, especially for cases with a large number of classes where the impact of the imbalance on minority classes is more severe. To this end, we compare the average test accuracy of the worst quartile of classes (worst 25 classes) for a ResNet-18 model trained on CIFAR-100 (Table 4). Our results indicate that the worst test accuracies improve in the presense of ReLU which could link to ReLU's impact on achieving the same train margin for all minority classes irrespective of the imbalance ratio.

| R | 5 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| MLP | 36.59 ± 0.19 | 23.66 ± 0.52 | 13.59 ± 0.83 | 6.73 ± 0.39 | 4.42 ± 0.65 |
| MLP + ReLU | 38.52 ± 0.24 | 27.49 ± 0.37 | 18.62 ± 0.72 | 10.41 ± 0.15 | 6.49 ± 0.35 |

**Table 3:** Worst class accuracy for ResNet-18 model trained using CIFAR-10 for different step imbalances.

| | | MLP | | MLP + ReLU | |
|---|---|---|---|---|---|
| | R | Step | Longtail | Step | Longtail |
| CIFAR-100 | 5 | 36.51 ± 0.12 | 39.70 ± 0.84 | 38.15 ± 0.52 | 41.35 ± 0.83 |
| | 10 | 23.12 ± 0.78 | 27.46 ± 0.18 | 31.46 ± 0.14 | 33.76 ± 0.19 |
| | 20 | 13.50 ± 0.86 | 23.30 ± 1.04 | 18.66 ± 1.05 | 25.52 ± 1.11 |
| | 50 | 6.72 ± 0.11 | 12.18 ± 0.18 | 10.44 ± 0.59 | 15.39± 0.19 |
| | 100 | 5.44 ± 0.37 | 7.13 ± 0.62 | 6.48 ± 0.47 | 8.99 ± 0.24 |

**Table 4:** Comparison of average test accuracies for the worst 25 classes for ResNet34 trained on CIFAR-100. For each setup, the test accuracies are evaluated over 5 runs. In all cases, we can see that the addition of ReLU helps improve the worst class accuracies.

### E.8.1 Impact of batch-binding on generalization

**Table 5:** NCC test accuracy for Batch-Binding

| | Step | | Long-tail | |
|---|---|---|---|---|
| Ratio ($R$) | Reshuffling | Reshuffling + Batch-Binding | Reshuffling | Reshuffling + Batch-Binding |
| 10 | 83.31 ± 0.65% | 83.27 ± 0.27% | 85.04 ± 0.28% | 85.03 ± 0.57% |
| 100 | 64.07 ± 1.79% | 64.18 ± 1.25% | 67.75 ± 1.21% | 68.08 ± 0.72% |

To study the effect of binding examples on generalization, we consider the *Nearest Class Center* (NCC) classifier. For this, each test example is assigned the label of the class center $c$ learned during training that is closest to it (in Euclidean distance). We consider a simple setup of training a ResNet-18 model on CIFAR10 under imbalances $R = 10, 100$ both with and without the binding examples. Models are trained with a batch size of 128 for 350 epochs. We ran the experiments with a smaller batch size to increase the number of back-propagation steps as adding data augmentations slows down convergence to OF geometry . We start with a learning rate of 0.1, which is reduced by a factor of 10 on epochs 250 and 300. Weight decay is set to $5 \times 10^{-4}$. In addition, we apply data augmentation as it is common practice when considering generalization and test accuracy. In particular, rather than simply adding horizontally flipped images (as described in Sec. 3) we allow for generic augmentations that include simple horizontal or vertical flips and random cropping with a probability of 0.5. The NCC test accuracy was measured across 5 versions of the experiment with the $k$ binding examples sampled randomly every time and 5 versions without any batch-binding. The results for NCC balanced test accuracies are provided in Table 5. While making definitive conclusions regarding the impact of the embeddings geometries and binding examples on generalization requires

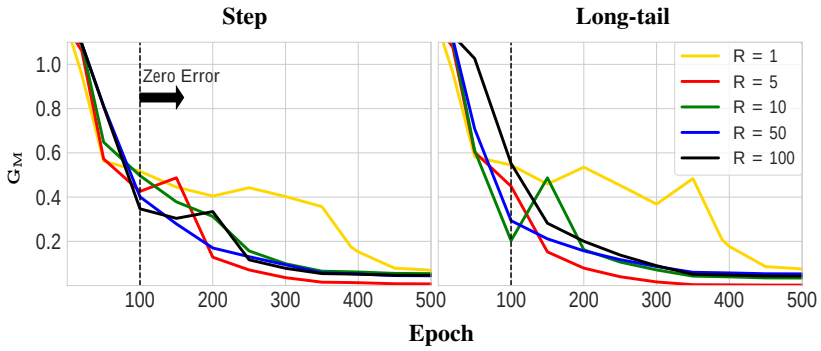


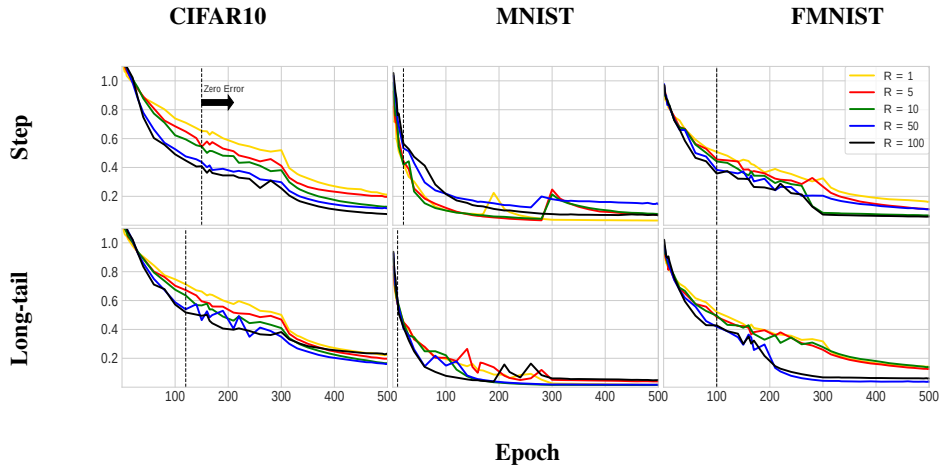

**Figure 22:** Convergence of $\mathbf{G_M}$ to the OF geometry for a ViT model trained on CIFAR10 under Step and Long-tail imbalance.

further investigation, this preliminary investigation suggests that batch-binding does *not* negatively impact NCC test accuracy.

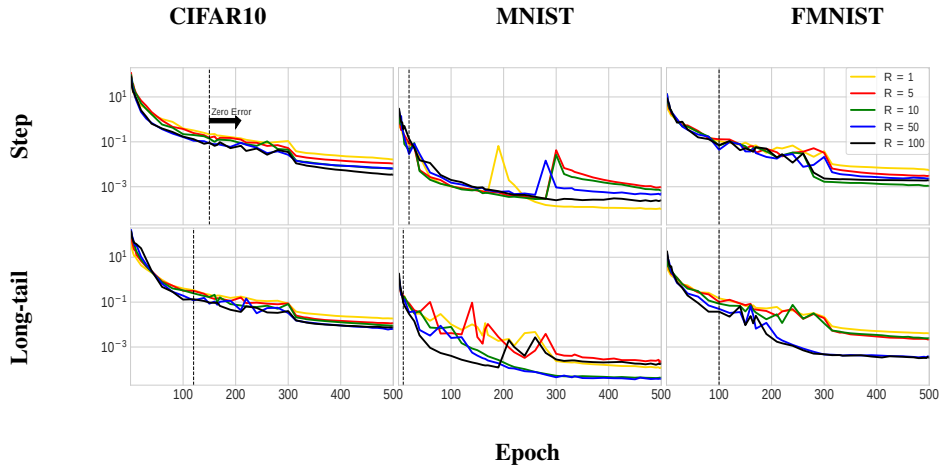
Finally, Fig. 17 shows convergence of embeddings geometry to OF for the same experiments. As expected, the convergence is slightly slower in this case due to the inclusion of data augmentation (random crops and flips).

### E.8.2 Experiments with additional architectures

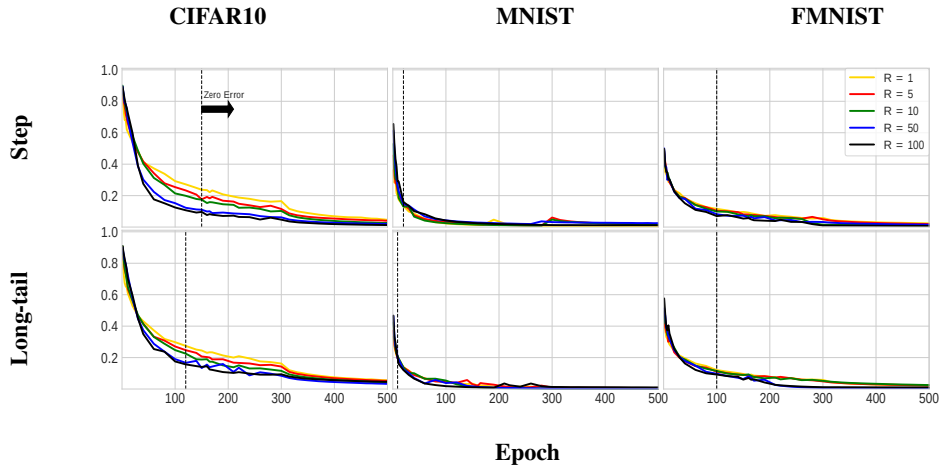
In order to ensure that our results are not impacted by the specifics of CNN structure of ResNet architecture, we perform a series of experiments on DenseNet-40 models and ViT Dosovitskiy et al. (2021). For the ViT models, we used a 2 layer vision transformer models with 10 attention heads for the CIFAR-10 dataset with exponential and step imbalances. Fig. 22 and Fig. 23 illustrate how the feature geometry converges to OF consistently for both networks, further solidifying our findings and indicating that given a large enough model, assumptions such that $\text{UFM}_+$ hold for classification tasks regardless of network architecture.




**(a)** $\mathbf{G}_{\mathrm{M}}$ Convergence to OF




**(b)** Neural Collapse




**(c)** Angles between class-means

**Figure 23:** Geometric convergence of embeddings (as based on Def. 1, Def. 2, and Def. 3) of DenseNet-40 trained with batch-binding. We measure: **(a)** $\mathbf{G}_{\mathrm{M}}$, **(b)** NC, and **(c)** $\mathrm{Ave}_{c \neq c'}\ \mathrm{sim}(c; c')$ as defined in text.