S1. Datasets

Real-world imbalanced datasets lack the controlability needed to study different levels of imbalance systematically. Therefore, we initially employed a highly controllable artificial dataset for preliminary experiments, only to then validate the results on real-world medical datasets that naturally exhibit binary distributions and class imbalance.

The artificial datasets were derived from the iNaturalist21 dataset [35]. We split this dataset into binary subsets based on taxonomic ranks and sub-sample them with varying levels of class imbalance. After identifying representation space issues and developing solutions using the controlled datasets, we validated our approach on four real-world medical datasets with natural binary class distributions: PneumoniaMNIST and BreastMNIST from Medical MNIST [40], a cardiac dataset from the UK Biobank [29] and FracAtlas [1].

S1.1. Artificially imbalanced datasets (iNat21)

Dataset selection is critical in (supervised) contrastive learning, as class semantics directly influence the learning process. High intra-class similarity (intra-class homogeneity) enhances the learning of discriminative features within classes, while low inter-class similarity (inter-class heterogeneity) aids in distinguishing between classes [32, 38]. Tsai et al. [33] emphasized the need for latent classes to embody task-relevant information within the training data.

To cover these effects, we select subsets of the iNaturalist 2021 (iNat21) dataset with different levels of homogeneity in and between classes. We determine semantic homogeneity between classes by using the hierarchical taxonomy, measuring class distances by steps in the taxonomy tree. Within-class heterogeneity is assessed based on the number of subspecies, taxonomic rank, and visual similarities of species, habitats, and backgrounds.

Based on these criteria, we selected three class categories:

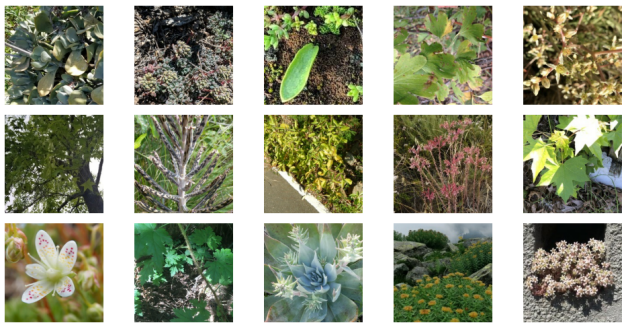
Dataset	Intra-class	Inter-class
Plants	Mixed*	Heterogeneous
Insects	Homogeneous	Homogeneous
Animals	Heterogeneous	Heterogeneous

*One class homogeneous, one class heterogeneous

S1.1.1. Plants dataset (asymmetric)

The dataset presents a clear contrast between its homogeneous and heterogeneous classes, marked by both inner-class characteristics and high between-class heterogeneity. Within this dataset, the Quercus genus, categorized under the taxonomy *Plantae* → *Tracheophyta* → *Magnoliopsida* → *Fagales* → *Fagaceae* → *Quercus* (Fig. S1a), represents a homogeneous class with 11,785 instances across 43 species.

(a) The homogeneous genus Quercus (oak)



(b) The heterogeneous Saxifragales order

Figure S1. The plants dataset illustrated by the homogeneous Quercus genus with its visually similar leaves and trees, and contrasting with the Saxifragales order, which exhibits high inner-class heterogeneity with a diverse array of plant forms from flowers to cacti, bushes, and trees.

This class is characterized by low inner-class heterogeneity, exhibiting minimal variance within the class, with visually similar leaves and trees.

In contrast, the axifragales order, following the taxonomy *Plantae* → *Tracheophyta* → *Magnoliopsida* → *Saxifragales* (Fig. S1b), serves as the heterogeneous class with 21,641 instances spanning 82 species. This class encompasses a wide variety of plant forms, including trees, shrubs, herbs, succulents, and aquatic plants, contributing to its high inner-class heterogeneity.

The distance between classes in the taxonomy tree is small, as Saxifragales is an order and thus two levels higher in the hierarchy than Quercus, a genus. This contrasts with the significant differences in class diversity and characteristics, emphasizing the dataset's asymmetry.

S1.1.2. Insects dataset (homogeneous)

This dataset comprises two closely related and homogeneous classes, Apidae (bees, Fig. S2a) and Vespidae (wasps, Fig. S2b). These classes are neighbors in the taxonomy tree with a branch distance of two, both belonging to the hierarchy level of family. They display similarities in species count, sample numbers, and visual characteristics, including consistent backgrounds in photography.

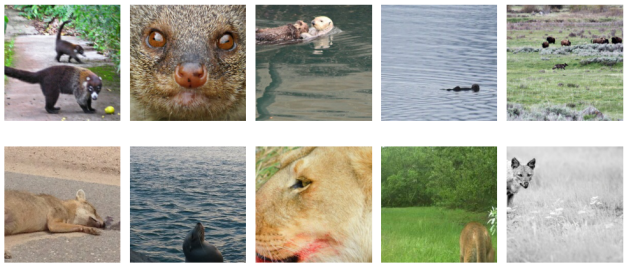
The Apidae family, which consists mainly of bees, is represented by 11,740 samples spanning 38 species. The

(a) Apidae class (bees)

(b) Vespidae class (wasps)

Figure S2. Representative images from the insects (homogeneous) dataset showcasing the two closely related classes, Apidae (bees) and Vespidae (wasps), exemplifying the dataset's homogeneity. Both classes demonstrate consistent visual characteristics, high intra-class homogeneity, and maintain a short taxonomic branch distance, underlining their similarities while retaining distinct biological traits.

(a) Artiodactyla class



(b) Carnivora order

Figure S3. Representative images from the animals (heterogeneous) dataset illustrating heterogeneity through examples from the diverse Artiodactyla class and the Carnivora order. The Artiodactyla exemplify inner-class diversity with species ranging from dolphins to giraffes and bovines, while the Carnivora, showing a similar diversity, includes species such as lions, ferrets, and sea lions. These images underscore the dataset's broad spectrum of biological diversity.

Vespidae family comprises 9,929 samples distributed across 42 species. Both families share a common taxonomic hierarchy, underlining their similarities while retaining distinct biological traits.

S1.1.3. Mammals dataset (heterogeneous)

This dataset focuses on two highly diverse classes of mammals, Artiodactyla (Fig. S3a) and Carnivora (Fig. S3b), both of which demonstrate significant inner-class and between-class heterogeneity. The Artiodactyla order, classified under Animalia, comprises 15,917 samples across 54 species. This group includes a wide range of species, such as deer, antelopes, bovines, dolphins, and giraffes, each with distinct morphological traits.

Similarly, the Carnivora order, contains 15,360 samples distributed among 55 species. This class encompasses predators and omnivores like bears, felines, canines, ferrets, and sea lions.

Artiodactyla species differ significantly from those in Carnivora, living in different ecological habitats and exhibiting a wide range of physical characteristics, highlighting the dataset's high between-class heterogeneity.

S1.2. Dataset splits

We sub-sampled our datasets to enable training across any split ranging from 1% to 99% for both classes while maintaining a constant total sample size across all experiments. To achieve this, we initially downsampled the more populous class to match the size of the smaller class before artificially imbalancing the two (see Tab. S1 for exact numbers).

S1.3. Medical datasets

S1.3.1. UK Biobank cardiac data

Our first medical dataset originates from the UK Biobank, a comprehensive biomedical database containing genetic and health data from over 500,000 UK individuals [29]. We used short-axis cardiovascular magnetic resonance (CMR) imaging data, originally comprising 46,656 subjects, each with a 4D MRI image stack. For our experiments, we utilized the middle slice of three time-points (End-Systolic, Mid-Systolic, and End-Diastolic) from each stack, which we encoded as an image's three channels. This dataset features class imbalances of 0.035 for infarction vs. rest and 0.086 for coronary artery disease (CAD) vs. rest. The labels were generated using the hospital admission ICD codes of the patients and include both past and future diagnoses. This was done to account for the fact that many cardiovascular diseases go undiagnosed for years until a severe event brings the patient into the hospital [9, 27, 34].

Imbalance Ratio	Total Samples	1% : 99%	5% : 95%	50% : 50%
Heterogeneous Dataset	14,577	145 : 14,432	728 : 13,849	7,289 : 7,289
Homogeneous Dataset	9,438	94 : 9,344	471 : 8,967	4,719 : 4,719
Asymmetric Dataset	11,197	111 : 11,086	559 : 10,638	5,599 : 5,599

Table S1. Distribution of samples across various levels of dataset imbalance. The table provides the count of samples for both classes in each scenario for heterogeneous, homogeneous, and asymmetric datasets. Test and validation sets are always balanced, ensuring valid comparisons between the splits.

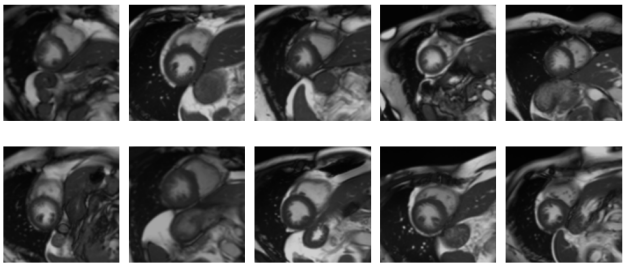


Figure S4. UKBB Cardiac

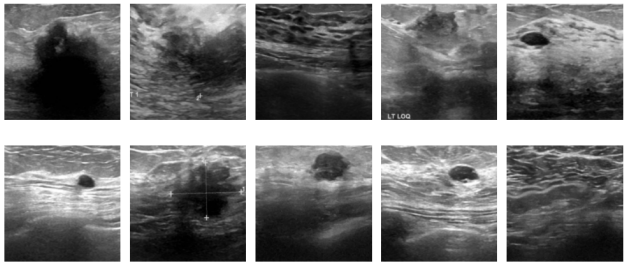


Figure S6. BreastMNIST

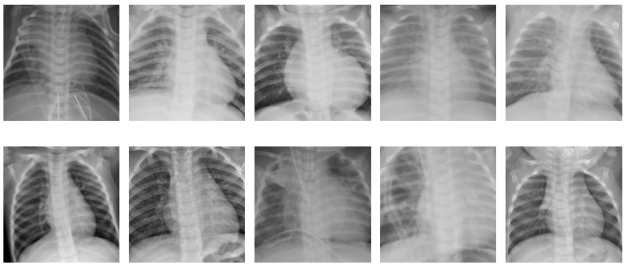


Figure S5. PneumoniaMNIST

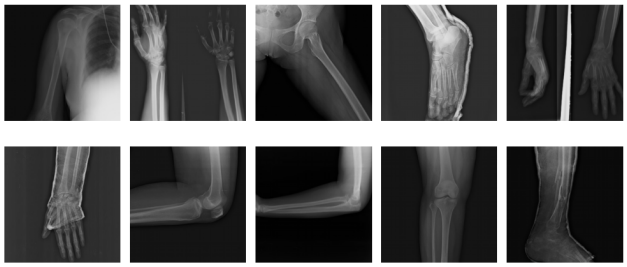


Figure S7. FracAtlas Dataset

S1.3.2. MedMNIST data

MedMNIST[40] provides standardized datasets for biomedical image classification with multiple size options: 28 (MNIST-Like), 64, 128, and 224 pixels. We chose the 224-pixel size and selected two datasets that naturally exhibit binary distributions. We use the original train, test and validation splits.

PneumoniaMNIST Derived from pediatric chest X-ray images, this dataset is used for binary classification of pneumonia with a class imbalance of 0.35 (positive) vs. 0.65 (negative). See Fig. S5 for some example images.

BreastMNIST Sourced from breast ultrasound images, this dataset categorizes images into normal and benign (grouped as positive) and malignant (negative) with an imbalance of 0.368 (positive) vs. 0.632 (negative). See Fig. S6 for some example images.

S1.3.3. FracAtlas Dataset

The FracAtlas dataset is a collection of medical imaging data focusing on bone fractures, published in Nature Scientific Data [1]. It includes 4,024 X-ray images annotated by medical professionals, covering different types of fractures across multiple anatomical locations such as the femur, tibia, humerus, radius, and others. The dataset features a class imbalance representative of clinical settings, with approximately 0.21 (fractured) vs. 0.79 (non-fractured), reflecting the lower proportion of fracture cases compared to normal cases typically seen in clinical practice. We use a 80%:10%:10% train, validation, and test splits. See Fig. S7 for some example images.

S2. Experimental setup

S2.1. Contrastive pre-training details

All experiments in the main paper were conducted using a ResNet-50 backbone [11]. The augmentations and linear projection head were adapted from the original SimCLR pa-

per with a projection dimension of 128 [5]. We set the batch size to 256, ensuring that at least one anchor of the minority class is included on average even at 1% imbalance ratio. Training was conducted for 350 epochs.

For optimization, we used stochastic gradient descent (SGD) with a momentum of 0.9 and a weight decay of 1×10^{-4}. A cosine annealing learning rate scheduler with a 10-epoch warm-up period [24] was utilized, starting with a learning rate of 0.00625 and warming up to 0.0625. We use a temperature value of 0.07. To ensure fairness, all approaches were trained for the same number of epochs and with the same backbone architecture. For all medical datasets, the training epochs were reduced to 250.

S2.1.1. Supervision in minority loss

Training details are consistent with the setup described above. The only modification is the loss function, as described in Sec. 4.2.

S2.1.2. Supervised prototype loss

Training details follow the same setup outlined above. The only difference is the loss function, detailed in Sec. 4.3.

S2.2. Weighted cross-entropy training details

As a baseline comparison, we employed weighted cross-entropy to counteract class imbalance effects. The model was optimized using Adam [18] with an initial learning rate of 1×10^{-4} and no dropout or weight decay applied. The weight used for each class was the inverse of its frequency.

S2.3. Evaluation details

To evaluate the quality of the learned representations, a linear probing protocol using a single linear layer was followed [5, 6, 17]. All pre-trained encoder weights were frozen and the linear head was trained for 50 epochs using only resize and center crop augmentations. A subset of 1% of the balanced pretraining dataset was used for linear probing along with a constant learning rate of $3 \times 10^{-4}$and the SGD optimizer with momentum of 0.9 and weight decay of 1×10^{-4}.

S2.4. Augmentations

S2.4.1. iNat21

For the iNat21 dataset, we used standard SimCLR transformations [5]. These include random resized cropping to 224×224 pixels and random horizontal flipping with a probability of 0.5. Additionally, we applied color jittering with a probability of 0.8 and random grayscaling with a probability of 0.2. Finally, we applied z-normalization to the images.

S2.4.2. MedMNIST & FracAtlas

For the MedMNIST and FracAtlas datasets, we applied augmentations specifically designed for grayscale medical images. Single-channel images were replicated across three channels. Subsequently, images were randomly cropped to a scale range of 25% to 100% of the original image area, with an aspect ratio range from 0.75 to 1.33, and then resized to the target size. Random horizontal flipping was applied with a probability of $p = 0.5$, along with selective color jittering to adjust brightness and contrast within a range of $\pm 15\%$ and a probability of $p = 0.8$. Finally, images were z-normalized as part of the transformation process.

S2.4.3. UKBB Cardiac

For the UKBB cardiac dataset, we used a combination of random horizontal flipping with a probability of $p = 0.5$ and random rotations up to 45 degrees. Color adjustments were applied to jitter brightness, contrast, and saturation within a range of $\pm 50\%$ and $p = 0.8$. Additionally, images were randomly resized with a scale range of 20% to 100% of the original image size, cropped to 128 pixels, and finally z-normalized.

S2.4.4. Evaluation transforms

For evaluation purposes, images were first resized to 256 pixels and then center-cropped to 224 pixels and z-normalized to maintain a consistent aspect ratio and size across all datasets.

S3. Baselines

S3.1. k-Positive Contrastive Learning (KCL)

In the k-Positive Contrastive Learning (KCL) method, we draw k instances from the same class to form the positive sample set. While the original paper by Kang *et al*. [15] sets $k = 6$, we also benchmarked with $k = 3$ due to the pronounced class imbalances in our dataset. For strong imbalances, in some batches, there are not enough positive samples for the majority class, averaging only $\lceil 2.56 \rceil = 3$. As demonstrated in the results section, we find that $k = 3$ is more effective for heavy imbalances in the binary case. We implemented the KCL loss directly in our pipeline and used the same hyperparameters as described above S2.

Although KCL is only briefly described in the appendix and named differently, it was also mentioned in the original Supervised Contrastive Learning (SupCon) paper [17].

S3.2. Targeted Supervised Contrastive Learning for Long-Tailed Recognition (TSC)

TSC extends KCL by introducing class prototypes. Instead of using the MoCo implementation from the authors' repository, we implemented the described loss within our SupCon framework for better comparability with other SupCon variations. We set the hyperparameter $\lambda = 1$ which weights the contribution of the prototypes to the total loss. Lambda is not specified in the original paper but was inferred from

the authors' code. The remaining hyperparameters were set as above. We tested both $k = 3$ and $k = 6$ in our experiments.

S3.3. Balanced Contrastive Learning for Long-Tailed Visual Recognition (BCL)

While TSC learns targets without explicit class semantics, BCL leverages class prototypes as additional samples. The BCL framework consists of a classification branch and a balanced contrastive learning branch, sharing a common backbone. The classifier weights are transformed by an MLP to serve as prototypes. We standardized the data augmentations with those used in other baselines for a fair comparison. The learning rate, batch size, and other training hyperparameters are identical to those described above.

S3.4. Subclass-Balancing Contrastive Learning for Long-Tailed Recognition (SBC)

We utilized the original authors' code for SBC but replaced the backbone with the same ResNet architecture used in all our experiments. The original class imbalance was maintained as the imbalance factor. We removed the warm-up period during which only SupCon is applied, as our experiments have shown that SupCon collapses for our data. Clusters were updated every 10 epochs, as suggested by the authors' code. We employed the "train rule rank" with a ranking temperature of 0.2 and used $grama = 0.25$ (grama in their code is called called β in their paper [14]), following the authors' recommendations. The rest of the hyperparameters are as described previously.

S3.5. Parametric contrastive learning baseline

Paco is a MoCo-based [12] strategy which we did not re-implement to SimCLR, considering momentum is a crucial component of the loss function. The parametric contrastive learning [7] baseline follows the hyperparameter suggestions of the original paper, setting the alpha parameter (α) to 0.05, the beta (β) and gamma (γ) parameters, which control the weighting of various losses, were both set to 1.0. Weight decay was set to 1×10^{-4}. The learning rate for this baseline was set at 0.0625. We used a MoCo-t temperature of 0.2, the MoCo queue size (MoCo-k) of 8192, and a MoCo embedding dimension (MoCo-dim) of 128. The momentum for the moving average encoder (MoCo-m) was set to 0.999.

S4. Additional baselines on traditional data imbalance strategies

We also evaluated several non-contrastive methods for mitigating data imbalance: majority-class undersampling, minority-class oversampling, and focal loss [23]. As shown in Tab. S2, these methods consistently underperform compared to both weighted cross-entropy and our proposed SupCon-based solutions. We conjecture that the extreme imbalance in our scenarios contributes to these results. At a 1% minority class ratio, undersampling yields only 188–249 training samples (spread across 80–125 species per dataset), lacking variability, while oversampling repeats the minority class up to 99% of the time, leading to overfitting.

	P 5%	**P 1%**	**I 5%**	**I 1%**	**A 5%**	**A 1%**
Focal	53.7	51.8	59.7	54.2	57.5	56.9
Oversample	59.7	50.0	59.0	52.1	58.8	57.9
Undersample	59.8	58.7	55.0	52.9	57.7	51.0

Table S2. P = plants, I = insects, A = animals

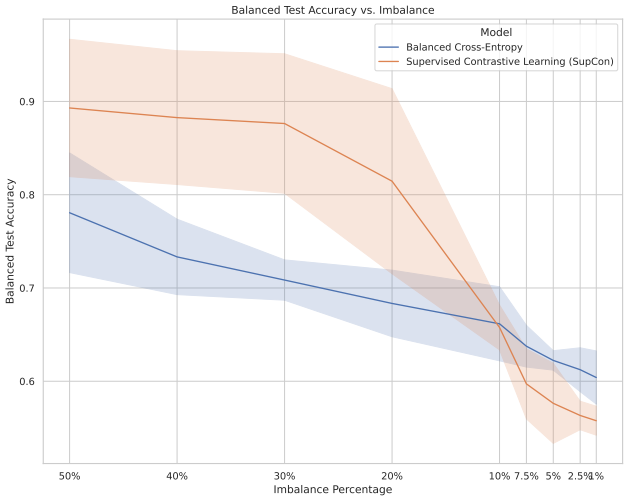


Figure S8. Balanced test performance averaged across three datasets - plants, insects, and animals - comparing SupCon and weighted cross-entropy under varying levels of dataset imbalance.

S5. Representation collapse during imbalanced binary supervised contrastive learning

Fig. S8 illustrates the balanced test performance averaged over the three datasets - animals, insects, and plants - for both SupCon and weighted cross-entropy (CE) across varying levels of dataset imbalance. We observe that SupCon consistently outperforms CE when the imbalance is low, indicating its superiority in balanced or slightly imbalanced scenarios. As the imbalance increases, a transition point emerges between 10% and 7.5% imbalance percentages, where the performance of SupCon is equal to that of CE. Beyond this point, SupCon's balanced test accuracy declines more sharply than CE's. In extreme imbalance conditions (e.g., 5%, 2.5%, and 1%) CE outperforms SupCon. These findings suggest that SupCon is highly effective in moderate imbalance conditions but struggles with extreme imbalance which is common to real world medical data.

S6. Representation space analysis on insects and animals datasets

Similarly to the results in the main paper (see Fig. 4) SupCon exhibits a representation space collapses at high data imbalances on the insects and animals datasets. Despite the canonical SAD and CAD metrics being low, SAA and CAC correctly identify the collapse. We also see an indication in the elevated SAA and CAC values that the collapse of the insects dataset at 5% imbalance was not quite as extreme as for the plants and animals datasets (62.6% accuracy vs 56.2% and 54.4%). This trend is also visible but much less pronounced in SAD and CAD.

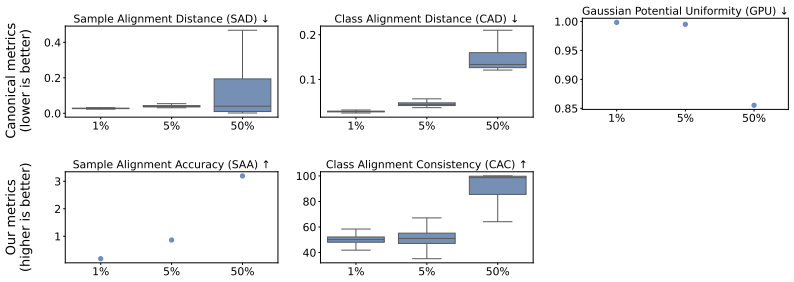


Figure S9. Analysis of SupCon's representation space learned from the insects dataset.

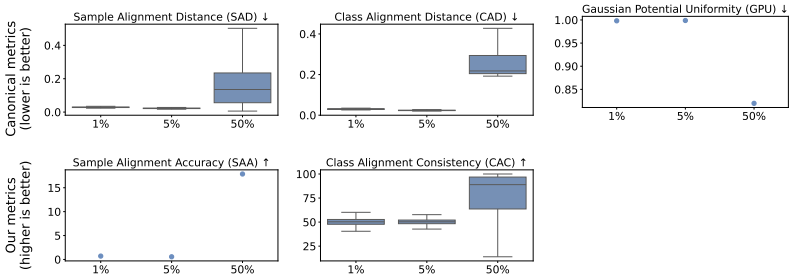


Figure S10. Analysis of SupCon's representation space learned from the animals dataset.

S7. Proving supervised contrastive representation collapse for binary class imbalances

In the following, we demonstrate that in SupCon (see Equation 4.1), the gradient of the output dimension can be effectively limited (upper bounded) by the count of positives associated with that sample. Consequently, an increase in the number of positives correlates with a decrease in the gradient magnitude.

Model	Dataset	Mean Cosine Similarity	Standard Deviation
Randomly Initialized ResNet50	Natural Images (Seed 1)	0.9983	0.0012
Randomly Initialized ResNet50	Natural Images (Seed 2)	0.9986	0.0010
Randomly Initialized ResNet50	Random Images	0.9979	0.0010
Randomly Initialized ResNet50	Pattern and Color Images	0.9989	0.0027
Pretrained IMAGENET1K_V2	Natural Images	0.0856	0.0635

Table S3. Initial Embedding Similarity in Randomly Initialized ResNet50

Initial behavior of randomly initialized Resnet50 For our analysis, we make assumptions regarding the initial state of the encoder output. At the start of training, the ResNet50 base encoder model [11] is initialized with random weights. Liange et al. [22] have empirically shown that uninitialized ResNet models tend to map their inputs to almost identical vectors, with a cosine similarity exceeding 0.99. We confirm their findings in our own empirical study (Tab. S3).

In our study, we observed that regardless of the input type—be it natural images, random images, or artificially dissimilar images (such as inverted patterns and colors)—the randomly initialized model consistently mapped these diverse inputs to remarkably similar output embeddings. Based on these observations, we propose the following lemma:

Lemma S7.1 *Let $z_i, z_k \in \mathcal{S}^{128}$ be two projections of an uninitialized ResNet50 model and an uninitialized Projection layer. Then, For a small $\varepsilon \in \mathbb{R}$, $\|z_i - z_k\| \leq \varepsilon$*

Let $\boldsymbol{w}_i$ denote the projection network output before normalization, i.e., $\boldsymbol{z}_i = \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}$ [17, p. 15]. In our analysis, we focus on w_i rather than p_i since when w_i is small, even a minor modification followed by normalization results in a proportionally larger change. Consequently, the gradient magnitude for smaller values of w_i is amplified.
$A(i) \equiv I \setminus \{i\}$ is the set of all indices without the anchor i in the multi-viewed batch. $P(i) \equiv \{p \in A(i) : \tilde{y}_p = \tilde{y}_i\}$ is the set of indices of all positives in the multi-viewed batch distinct from i. $N(i) \equiv A(i) \setminus P(i)$ is the set of indices of all negatives in the multi-viewed batch.

Following Khosla *et al.* [17, p. 16], the gradient of the supervised loss in relation to $\boldsymbol{w}_i$, and restricted to $P(i)$ or $N(i)$ is

$$\left.\frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i}\right|_{P(i)} = \frac{1}{\tau||\boldsymbol{w}_i||} \sum_{p \in P(i)} (\boldsymbol{z}_p - (\boldsymbol{z}_i \cdot \boldsymbol{z}_p)\boldsymbol{z}_i)(P_{ip} - X_{ip}) \tag{13}$$

$$\left.\frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i}\right|_{N(i)} = \frac{1}{\tau||\boldsymbol{w}_i||} \sum_{n \in N(i)} (\boldsymbol{z}_n - (\boldsymbol{z}_i \cdot \boldsymbol{z}_n)\boldsymbol{z}_i)P_{in} \tag{14}$$

Summing Eqs. 13 and 14 gives us the gradient of the supervised loss with respect to $\boldsymbol{w}_i$:

$$\frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i} = \left.\frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i}\right|_{P(i)} + \left.\frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i}\right|_{N(i)} \tag{15}$$

We further define following Khosla *et al.* [17, p. 16]:

$$P_{ix} \equiv \frac{\exp(\boldsymbol{z_i} \cdot \boldsymbol{z_x}/\tau)}{\sum_{a \in A(i)} \exp(\boldsymbol{z_i} \cdot \boldsymbol{z_a}/\tau)} \tag{16}$$

and

$$X_{ip} \equiv \frac{1}{|P(i)|} \tag{17}$$

Theorem S7.2 *Let us consider the context of Lemma S7.1, where we assume for a small $\varepsilon \in \mathbb{R}$, that $\|\boldsymbol{z}_i - \boldsymbol{z}_j\| \leq \varepsilon$. Furthermore, given that $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{S}^{128}$, we have $\|\boldsymbol{z}_i\| \cdot \|\boldsymbol{z}_j\| = 1$. Under these conditions, the following inequality holds for the size of the output gradients:*

$$\left\| \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{w}_i} \right\| \leq \frac{1}{\tau \|\boldsymbol{w}_i\|} (\varepsilon + \frac{1}{2}\varepsilon^2) \left((1 - \frac{|P(i)|}{|A(i)|}) \exp(-\varepsilon^2/\tau) + (\exp(\varepsilon^2/\tau) - 1) + (1 - \frac{|P(i)|}{|A(i)|})^2 \exp(\varepsilon^2/\tau) \right)$$

Proof

$$\left\| \left. \frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i} \right|_{P(i)} \right\| \leq \frac{1}{\tau \|\boldsymbol{w}_i\|} \sum_{p \in P(i)} \|\boldsymbol{z}_p - (\boldsymbol{z}_i \cdot \boldsymbol{z}_p)\boldsymbol{z}_i\| |P_{ip} - X_{ip}|$$

First we show:

$$|P_{ip} - X_{ip}| \leq \exp(-\varepsilon^2/\tau) \frac{|N(i)|}{|P(i)||A(i)|} + (\exp(\varepsilon^2/\tau) - 1) \frac{1}{|P(i)|}$$

with

$$\begin{aligned}
&\frac{\exp((1 - \varepsilon^2/2)/\tau)}{\exp((1 + \varepsilon^2/2)/\tau)} \frac{1}{|A(i)|} \\
&\leq \frac{\exp((1 - \|z_i - z_p\|^2/2)/\tau)}{\exp((1 + \varepsilon^2/2)/\tau)} \frac{1}{|A(i)|} && \text{Theorem S7.1} \\
&= \frac{\exp((1 - (z_i^2 - 2 z_i z_p + z_p^2)/2)/\tau)}{\exp((1 + \varepsilon^2/2)/\tau)} \frac{1}{|A(i)|} \\
&= \frac{\exp((1 - (\frac{1}{2} - z_i z_p + \frac{1}{2}))/\tau)}{\exp((1 + \varepsilon^2/2)/\tau)} \frac{1}{|A(i)|} \\
&= \frac{\exp((1 - (1 - z_i z_p))/\tau)}{\exp((1 + \varepsilon^2/2)/\tau)} \frac{1}{|A(i)|} \\
&= \frac{\exp((\boldsymbol{z}_i \cdot \boldsymbol{z}_p)/\tau)}{\exp((1 + \varepsilon^2/2)/\tau)} \frac{1}{|A(i)|} \\
&\leq \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p/\tau)}{\sum_{a \in A(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a/\tau)} \\
&\leq \frac{\exp((1 + \varepsilon^2/2)/\tau)}{\exp((1 - \varepsilon^2/2)/\tau)} \frac{1}{|A(i)|}
\end{aligned}$$

if $P_{ip} \geq X_{ip}$:

$$\begin{aligned}
|P_{ip} - X_{ip}| &= P_{ip} - X_{ip} \\
&= \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p/\tau)}{\sum_{a \in A(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a/\tau)} - \frac{1}{|P(i)|} \\
&\leq \frac{\exp((1 + \varepsilon^2/2)/\tau)}{\exp((1 - \varepsilon^2/2)/\tau)} \frac{1}{|A(i)|} - \frac{1}{|P(i)|} \\
&= \frac{\exp((1 + \varepsilon^2/2)/\tau)}{\exp((1 - \varepsilon^2/2)/\tau)} \frac{1}{|P(i)| + |N(i)|} - \frac{1}{|P(i)|} \\
&\leq \frac{\exp((1 + \varepsilon^2/2)/\tau)}{\exp((1 - \varepsilon^2/2)/\tau)} \frac{1}{|P(i)|} - \frac{1}{|P(i)|} \\
&= \exp(\varepsilon^2/\tau) \frac{1}{|P(i)|} - \frac{1}{|P(i)|} \\
&= (\exp(\varepsilon^2/\tau) - 1) \frac{1}{|P(i)|}
\end{aligned}$$

if $X_{ip} \geq P_{ip}$:

$$
\begin{aligned}
|P_{ip} - X_{ip}| &= X_{ip} - P_{ip} \\
&\leq \frac{1}{|P(i)|} - \frac{\exp((1-\varepsilon^2/2)/\tau)}{\exp((1+\varepsilon^2/2)/\tau)} \frac{1}{|A(i)|} \\
&= \frac{1}{|P(i)|} + \exp(-\varepsilon^2/\tau) \frac{-1}{|A(i)|} \\
&= \frac{1}{|P(i)|} + \exp(-\varepsilon^2/\tau) \left[\frac{-|P(i)|}{|P(i)||A(i)|} + \frac{|A(i)|}{|P(i)||A(i)|} - \frac{|A(i)|}{|P(i)||A(i)|} \right] \\
&= \frac{1}{|P(i)|} + \exp(-\varepsilon^2/\tau) \left[\frac{|A(i)| - |P(i)|}{|P(i)||A(i)|} - \frac{|A(i)|}{|P(i)||A(i)|} \right] \\
&= \frac{1}{|P(i)|} + \exp(-\varepsilon^2/\tau) \left[\frac{|A(i)| - |P(i)|}{|P(i)||A(i)|} - \frac{1}{|P(i)|} \right] \\
&= \exp(-\varepsilon^2/\tau) \frac{|N(i)|}{|A(i)||P(i)|} + (1 - \exp(-\varepsilon^2/\tau)) \frac{1}{|P(i)|} \\
&\leq \exp(-\varepsilon^2/\tau) \left(\frac{|N(i)|}{|A(i)||P(i)|} + (\exp(\varepsilon^2/\tau) - 1) \frac{1}{|P(i)|} \right)
\end{aligned}
$$

$$
\implies |P_{ip} - X_{ip}| \leq \exp(-\varepsilon^2/\tau) \left(\frac{|N(i)|}{|P(i)||A(i)|} + (\exp(\varepsilon^2/\tau) - 1) \frac{1}{|P(i)|} \right)
$$

$$
\begin{aligned}
\|\boldsymbol{z}_p - (\boldsymbol{z}_i \cdot \boldsymbol{z}_p)\boldsymbol{z}_i\| &= \|\boldsymbol{z}_p - \boldsymbol{z}_i + \boldsymbol{z}_i - (\boldsymbol{z}_i \cdot \boldsymbol{z}_p)\boldsymbol{z}_i\| \\
&\leq \|\boldsymbol{z}_p - \boldsymbol{z}_i\| + |1 - (\boldsymbol{z}_i \cdot \boldsymbol{z}_p)| \|\boldsymbol{z}_i\| \\
&\leq \varepsilon + |1 - \mathbf{z}_i \cdot \mathbf{z}_p| \\
&= \varepsilon + \left| 1 + \frac{1}{2}(\mathbf{z}_i - \mathbf{z}_p) \cdot (\mathbf{z}_i - \mathbf{z}_p) - \frac{1}{2}\mathbf{z}_i \cdot \mathbf{z}_i - \frac{1}{2}\mathbf{z}_p \cdot \mathbf{z}_p \right| \\
&\leq \varepsilon + \left| \frac{1}{2}(\mathbf{z}_i - \mathbf{z}_p) \cdot (\mathbf{z}_i - \mathbf{z}_p) \right| \\
&\leq \varepsilon + \frac{1}{2}\varepsilon^2
\end{aligned}
$$

$$
\begin{aligned}
\left\| \left. \frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i} \right|_{N(i)} \right\| &\leq \frac{1}{\tau \|\boldsymbol{w}_i\|} \sum_{n \in N(i)} \|\boldsymbol{z}_n - (\boldsymbol{z}_i \cdot \boldsymbol{z}_n)\boldsymbol{z}_i\| |P_{in}| \\
&\leq \frac{1}{\tau \|\boldsymbol{w}_i\|} \frac{|N(i)|}{|A(i)|} (\varepsilon + \frac{1}{2}\varepsilon^2) \exp(\varepsilon^2/\tau)
\end{aligned}
$$

Finally,

$$
\begin{aligned}
\left\| \frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i} \right\| &\leq \left\| \left. \frac{\partial \mathcal{L}_i^{sup}}{\partial \boldsymbol{w}_i} \right|_{P(i)} \right\| + \left\| \left. \frac{\partial \mathcal{L}^{sup}}{\partial \boldsymbol{w}_i} \right|_{N(i)} \right\| \\
&\leq \frac{1}{\tau \|\boldsymbol{w}_i\|} \left(\sum_{p \in P(i)} \|\boldsymbol{z}_p - (\boldsymbol{z}_i \cdot \boldsymbol{z}_p)\boldsymbol{z}_i\| |P_{ip} - X_{ip}| + \sum_{n \in N(i)} \|\boldsymbol{z}_n - (\boldsymbol{z}_i \cdot \boldsymbol{z}_n)\boldsymbol{z}_i\| |P_{in}| \right) \\
&\leq \frac{1}{\tau \|\boldsymbol{w}_i\|} \left(\sum_{p \in P(i)} (\varepsilon + \frac{1}{2}\varepsilon^2) \left(\exp(-\varepsilon^2/\tau)(\frac{|N(i)|}{|A(i)||P(i)|} + (\exp(\varepsilon^2/\tau) - 1)\frac{1}{|P(i)|}) \right) \right. \\
&\quad \left. + (\varepsilon + \frac{1}{2}\varepsilon^2)\frac{|N(i)|}{|A(i)|}\exp(\varepsilon^2/\tau) \right) \\
&\leq \frac{1}{\tau \|\boldsymbol{w}_i\|} \left(|P(i)|(\varepsilon + \frac{1}{2}\varepsilon^2) \left(\exp(-\varepsilon^2/\tau)(\frac{|N(i)|}{|P(i)||A(i)|} + (\exp(\varepsilon^2/\tau) - 1)\frac{1}{|P(i)|}) \right) \right. \\
&\quad \left. + (\varepsilon + \frac{1}{2}\varepsilon^2)\frac{|N(i)|}{|A(i)|}\exp(\varepsilon^2/\tau) \right) \\
&= \frac{1}{\tau \|\boldsymbol{w}_i\|}(\varepsilon + \frac{1}{2}\varepsilon^2) \left(\exp(-\varepsilon^2/\tau)(\frac{|N(i)|}{|A(i)|} + (\exp(\varepsilon^2/\tau) - 1)\frac{|P(i)|}{|P(i)|}) \right) \\
&\quad + \frac{1}{\tau \|\boldsymbol{w}_i\|}(\varepsilon + \frac{1}{2}\varepsilon^2) \left(\frac{|N(i)|}{|A(i)|}\exp(\varepsilon^2/\tau) \right) \\
&= \frac{1}{\tau \|\boldsymbol{w}_i\|}(\varepsilon + \frac{1}{2}\varepsilon^2) \left(\frac{|N(i)|}{|A(i)|}\exp(-\varepsilon^2/\tau) + \exp(-\varepsilon^2/\tau)(\exp(\varepsilon^2/\tau) - 1) + \frac{|N(i)|}{|A(i)|}\exp(\varepsilon^2/\tau) \right) \\
&= \frac{1}{\tau \|\boldsymbol{w}_i\|}(\varepsilon + \frac{1}{2}\varepsilon^2) \left((1 - \frac{|P(i)|}{|A(i)|})\exp(-\varepsilon^2/\tau) + \exp(-\varepsilon^2/\tau)(\exp(\varepsilon^2/\tau) - 1) + (1 - \frac{|P(i)|}{|A(i)|})\exp(\varepsilon^2/\tau) \right)
\end{aligned}
$$

□

Based on the theorem's conclusions, we see that the gradients for the loss function in supervised contrastive learning are upper-bounded by the number of positive samples. In the context of severe class imbalances, the increment in the number of positive samples (from the majority class) may dominate the output vector. This dominance can constrain the gradient magnitudes, causing them to become too small to induce effective weight updates in the network.

S8. Ablations

S8.1. Ablations on temperature and batch size

Experiments on temperature reveal that both fixes are robust across a range of temperature settings, with optimal results observed for temperatures between 0.1 and 0.5. While low to medium temperatures do not alleviate the collapse in SupCon, very high temperatures can mitigate collapse issues in moderately imbalanced scenarios; however, this comes at a cost, resulting in an accuracy that is 19% lower than our proposed method. Fig. S11

Furthermore, unlike supervised contrastive learning in balanced multi-class datasets, we find that increasing batch sizes negatively affects performance. We attribute this degradation to the larger number of positive pairs per sample introduced by bigger batches, leading to collapse. A detailed theoretical justification of this phenomenon is provided in Sec. S7.

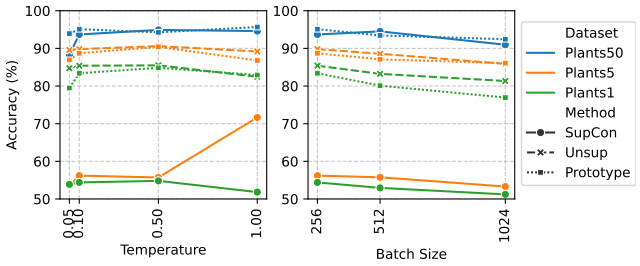


Figure S11. Ablation of our fixes and SupCon for different batch sizes and temperatures using the plants dataset. Our fixes are robust across a range of temperatures and batch sizes, though increasing batch size typically degrades performance slightly.

S8.2. Supervised minority ablation

In Tab. S4, we investigate how varying levels of supervision in the majority class impact performance on an imbalanced dataset, while keeping the supervision level fixed in the minority class (see Sec. 4.2). The study was conducted using the insects dataset composed of 5% minority samples and 95% majority samples. This shows that our strategy of full supervision in the minority and no supervision in the majority performs best in these strong imbalance scenarios. A notable drop in performance occurs between 5% and 1% supervision where the representations collapse. This ablation study is similar to KCL [15] under varying levels of K. The results are consistent with our KCL baselines as we find that a larger K in a batch is harmful for downstream utility. For a batch size of 256, 5% supervision already translates to $K = 12.8$.

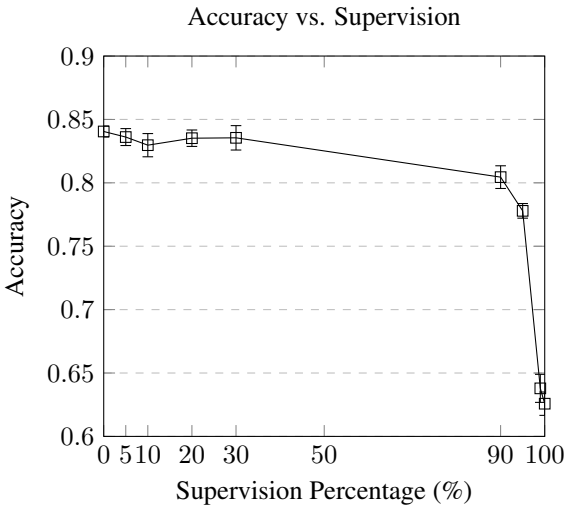


Figure S12. Increasing supervision in the majority class vs. balanced test accuracy.

Supervision θ	Accuracy (%)
0%	**84.05** $\pm$ 0.43
5%	83.61 $\pm$ 0.66
10%	82.97 $\pm$ 0.91
20%	83.52 $\pm$ 0.64
30%	83.55 $\pm$ 0.96
90%	80.45 $\pm$ 0.89
95%	77.79 $\pm$ 0.57
99%	63.79 $\pm$ 1.10
100%	62.58 $\pm$ 0.92

Table S4. For our Supervised Minority fix, we show the effects of increasing supervision in the majority class on the insects dataset with 5% imbalance. No amount of supervision in the majority class improves downstream performance.

S8.3. Supervised majority ablation

Majority Supervision (%)	**Label Imbalance**	
	5%	**1%**
10	67.44	55.93
50	54.72	53.11
100	52.16	51.58

Table S5. Test accuracy when supervision is applied to the majority class instead of the minority class, evaluated at different supervision levels on the Plants dataset with 5% and 1% imbalance.

We evaluated using SupCon loss for the majority class and NT-Xent loss for the minority class. With no supervision in the minority class, even mild supervision in the majority class failed to train effectively.

S9. UMAP Visualization

UMAP [25] is a dimensionality reduction technique that is widely used for visualizing high-dimensional data. We employ UMAP to visualize the embeddings of all three datasets - *plants*, *insects*, and *animals* - under varying levels of class imbalance, using unseen test data (see Figs. S13 to S15).

The UMAP visualizations of the SupCon embedding spaces corroborate our findings from the main paper: the embedding space collapses under strong class imbalances, resulting in diminished utility. Even in the balanced case, the two classes are not distinctly separated. It is important to note that UMAP represents pairwise distances in a relative manner, which can obscure the visualization of an embedding space collapsing to a single vector. The relative scaling in UMAP means that even minimal differences between embeddings can appear more pronounced, masking the extent of the collapse.

In contrast, the *Supervised Prototype Fix* and the *Supervised Minority Fix* methods exhibit clear class clusters and separation across all levels of imbalance. This observation aligns with our theoretical illustrations presented in Figure 3.

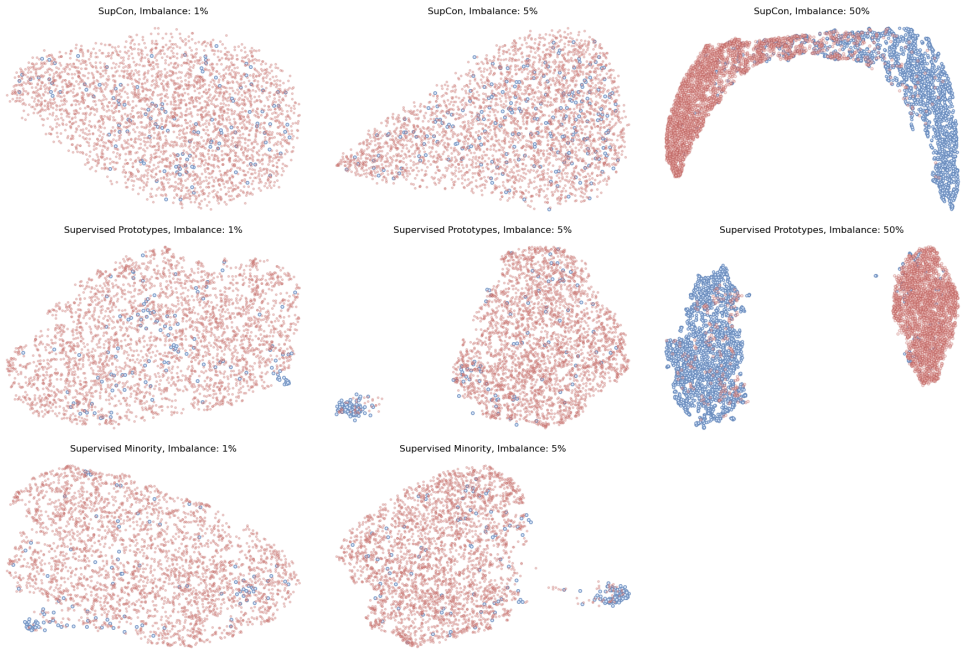


Figure S13. UMAP visualization of projection space after supervised pre-training on the plants dataset.

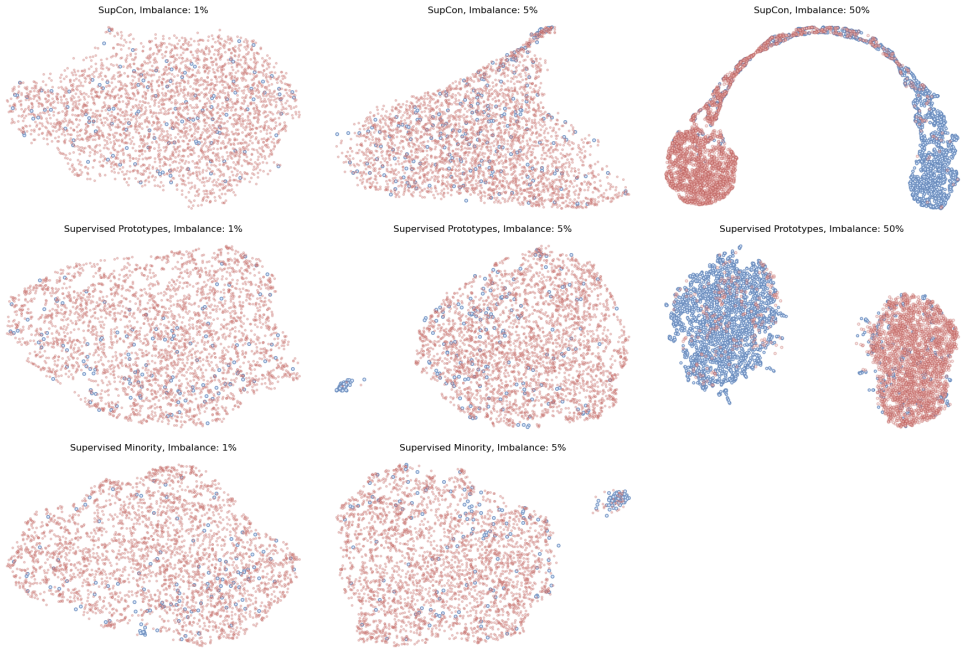


Figure S14. UMAP visualization of projection space after supervised pre-training on the insects dataset.

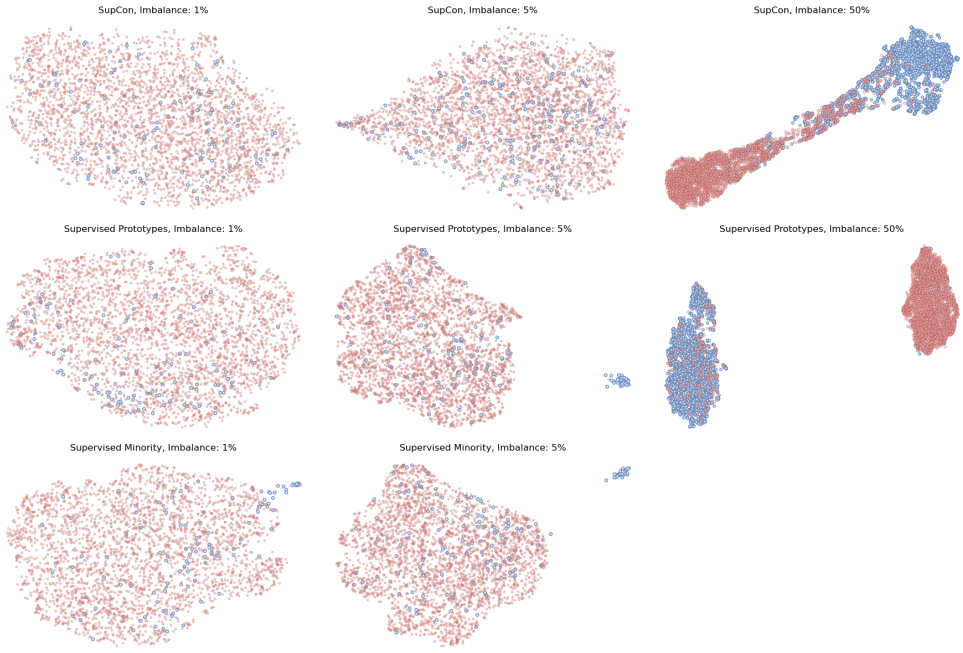


Figure S15. UMAP visualization of projection space after supervised pre-training on the animals dataset.