Supplementary- SpHOR: A Representation Learning Perspective on Open-set Recognition for Identifying Unknown Classes in Deep Neural Networks

Thiru Thillai Nadarasar Bahavan, Sachith Seneviratne, Saman Halgamuge
The University of Melbourne, Australia
bahavant@student.unimelb.edu.au, {sachith.seneviratne,saman.halgamuge}@unimelb.edu.au

1. Preliminaries

To further encourage the model to focus on these meaningful, class-specific attributes and prevent it from relying on shared features, we introduce an orthogonality loss. This loss function ensures that the feature embeddings for each class reside in distinct, linearly independent, and orthogonal subspaces. By enforcing this constraint, a potentially shared attribute (like a general feature on 'Axis 2') is effectively converted into a class-specific attribute, forcing the model to learn truly discriminative representations and thereby enhancing its ability to accurately classify both known and novel classes as Fig 1.

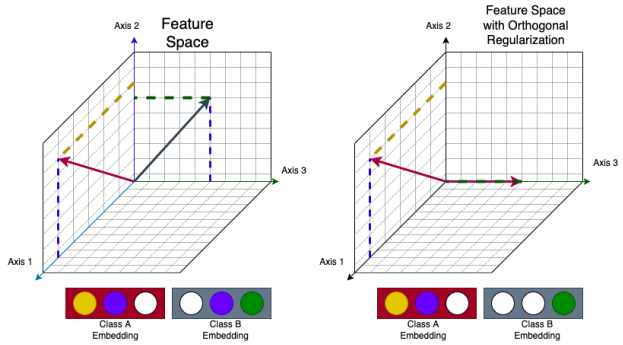


Figure 1. To encourage the model to learn class-specific attributes, avoiding shared features by ensuring that embeddings from different classes reside in distinct, linearly independent, and orthogonal subspaces, we implement an orthogonality loss. Axis 2 is a class-shared attribute which is converted to a class-specific attribute via orthogonalization loss.

2. Methods

2.1. Proof of Theorem 01

Suppose we consider a sample (x_i, y_i), then our loss is:

$$\mathcal{L}_{\text{vMFAL}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} S_{ij}\log P_{ij} \tag{1}$$

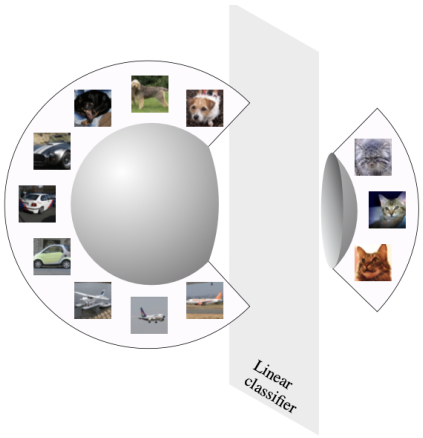


Figure 2. When classes are well-clustered, exhibiting both uniformity and alignment, they form hyperspherical clusters. In such hyperspheres, the clusters become linearly separable, often taking the shape of spherical caps. This property, however, does not generally hold in standard Euclidean spaces. This Figure was taken directly, from [19]

The per-sample loss is:

$$\mathcal{L}^{(i)}_{\text{vMFAL}} := -\sum_{j=1}^{C} S_{ij}\log P_{ij} \tag{2}$$

$$\mathbb{P}(y=j \mid \mathbf{z_i}; \{\boldsymbol{\mu}_k\}_{k=1}^{C}) = P_{ij} = \frac{e^{\left(z_i^\top \boldsymbol{\mu}_j/\tau\right)}}{\sum_{k\in C} e^{\left(z_i^\top \boldsymbol{\mu}_k/\tau\right)}}. \tag{3}$$

,

Where $\tilde{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iC}]$, and $S_{ij} = \frac{y_{ij}}{\sum_{k\in C} y_{ik}}$. Assume μ_j are fixed/constant with respect to z_i.

Theorem: *The first derivative of the loss $\mathcal{L}^{(i)}_{vMFAL}$ for a sample i is:*

$$\frac{\partial \mathcal{L}^{(i)}_{\text{vMFAL}}}{\partial z_i} = -\sum_{j\in C}[S_{ij} - P_{ij}]\left(\boldsymbol{\mu}_j/\tau\right) \tag{4}$$

From the definition,

$$\mathcal{L}^{(i)}_{\text{vMFAL}} = -\sum_{j\in C} S_{ij}\log(\frac{e^{\left(z_i^\top \boldsymbol{\mu}_j/\tau\right)}}{\sum_{k\in C} e^{\left(z_i^\top \boldsymbol{\mu}_k/\tau\right)}})$$

$$\frac{\partial \mathcal{L}^{(i)}_{\text{vMFAL}}}{\partial z_i} = -\sum_{j\in C} S_{ij}\frac{\partial}{\partial z_i}\log P_{ij}$$

From Lemma 2.1.1,

$$\frac{\partial \mathcal{L}^{(i)}_{\text{vMFAL}}}{\partial z_i} = -\frac{1}{\tau}\sum_{j\in C} S_{ij}\boldsymbol{\mu}_j + \frac{1}{\tau}\sum_{j\in C}(S_{ij}\sum_{k\in C} P_{ik}\boldsymbol{\mu}_k)$$

$$= -\frac{1}{\tau}\sum_{j\in C} S_{ij}\boldsymbol{\mu}_j + \frac{1}{\tau}\sum_{k\in C}(P_{ik}\boldsymbol{\mu}_k\sum_{j\in C} S_{ij})$$

Since $\sum_{j\in C} S_{ij} = 1$, by the definition

$$= \frac{-1}{\tau}\sum_{j\in C} S_{ij}\boldsymbol{\mu}_j - \frac{-1}{\tau}\sum_{k\in C} P_{ik}\boldsymbol{\mu}_k$$

$$= \frac{-1}{\tau}\sum_{j\in C}[S_{ij}\boldsymbol{\mu}_j - P_{ij}\boldsymbol{\mu}_j] = -\frac{1}{\tau}\sum_{j\in C}[S_{ij} - P_{ij}]\boldsymbol{\mu}_j$$

- When the model *underestimates* the similarity for class j (i.e., $S_{ij} > P_{ij}$), the gradient encourages z_i to move *toward* the class weight vector $\boldsymbol{\mu}_j$.
- Conversely, when the model *overestimates* the similarity for class j (i.e., $P_{ij} > S_{ij}$), the gradient pushes z_i *away from* $\boldsymbol{\mu}_j$.

2.1.1. Proof of Lemma 01

Below is a short, self-contained proof of the gradient of $\log P_{ij}$ with respect to z_i.

$$\log P_{ij} = \frac{z_i^\top \mu_j}{\tau} - \log\Big(\sum_{k\in C} e^{z_i^\top \mu_k/\tau}\Big)$$

Differentiate the first term:

$$\frac{\partial}{\partial z_i}\frac{z_i^\top \mu_j}{\tau} = \frac{\mu_j}{\tau}$$

Differentiate the second term:

$$\frac{\partial}{\partial z_i}\log\Big(\sum_{k\in C} e^{z_i^\top \mu_k/\tau}\Big) = \frac{1}{\sum_\ell e^{z_i^\top \mu_\ell/\tau}}\sum_{k\in C} e^{z_i^\top \mu_k/\tau}\frac{\mu_k}{\tau}$$

$$\frac{\partial}{\partial z_i}\log\Big(\sum_{k\in C} e^{z_i^\top \mu_k/\tau}\Big) = \frac{1}{\tau}\sum_{k\in C} P_{ik}\mu_k$$

Combine the two contributions:

$$\frac{\partial}{\partial z_i}\log P_{ij} = \frac{1}{\tau}\Big(\mu_j - \sum_{k\in C} P_{ik}\mu_k\Big)$$

2.2. Proof of Theorem 02

Our proposed loss promotes *Uniformity* and *Alignment* in our representations.

Suppose we consider a sample (x_i, y_i), then our loss is:

$$\mathcal{L}_{\text{vMFAL}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C} S_{ik}\log P_{ik} \tag{5}$$

$$\mathcal{L}_{\text{vMFAL}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C} S_{ik}\log(\frac{\frac{e^{\mu_k^T z_i}}{\tau}}{\sum_{k=1}^{C} e^{\frac{\mu_j^T z_i}{\tau}}})$$

Expanding the log terms and the exponents,

$$\mathcal{L}_{\text{vMFAL}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C}\left(S_{ik}\frac{\mu_k^T z_i}{\tau} - S_{ik}\log\sum_{j=1}^{C} e^{\frac{\mu_j^T z_i}{\tau}}\right)$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{C} S_{ik}\frac{\mu_k^T z_i}{\tau} - \sum_{k=1}^{C} S_{ik}\log\sum_{j=1}^{C} e^{\frac{\mu_j^T z_i}{\tau}}\right)$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{C} S_{ik}\frac{\mu_k^T z_i}{\tau} - \left(\sum_{k=1}^{C} S_{ik}\right)\log\sum_{j=1}^{C} e^{\frac{\mu_j^T z_i}{\tau}}\right)$$

$$\text{since } (\sum_{k\in C} S_{ik} = 1)$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C} S_{ik}\frac{\mu_k^T z_i}{\tau} + \frac{1}{N}\sum_{i=1}^{N}\log\sum_{j=1}^{C} e^{\frac{\mu_j^T z_i}{\tau}}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(-\sum_{k=1}^{C} S_{ik}\frac{\mu_k^T z_i}{\tau} + \log\sum_{j=1}^{C} e^{\frac{\mu_j^T z_i}{\tau}}\right)$$

Thus,

$$\mathcal{L}^{(i)}_{\text{vMFAL}} = \underbrace{-\frac{1}{\tau}\sum_{k=1}^{C} S_{ik}(z_i^\top \boldsymbol{\mu}_k)}_{\text{Alignment}} + \underbrace{\log\sum_{k=1}^{C}\exp\left(\frac{z_i^\top \boldsymbol{\mu}_k}{\tau}\right)}_{\text{Uniformity}} \tag{6}$$

Using the identity for unit-normed vectors

$$z_i^\top \mu_k = 1 - \frac{1}{2}\|z_i - \mu_k\|^2,$$

the alignment term becomes

$$-\frac{1}{\tau}\sum_{k=1}^{C} S_{ik}(z_i^\top \mu_k) = -\frac{1}{\tau} + \frac{1}{2\tau}\sum_{k=1}^{C} S_{ik}\|z_i - \mu_k\|^2,$$

and the uniformity term becomes

$$\log\sum_{k=1}^{C}\exp\left(\frac{z_i^\top \mu_k}{\tau}\right) = \frac{1}{\tau} + \log\sum_{k=1}^{C}\exp\left(-\frac{1}{2\tau}\|z_i - \mu_k\|^2\right).$$

2.3. Alignment–Uniformity Balancing

1. Alignment Term

For a (mixed) ambiguous target $S_{ik} \approx 1/C$ we have

$$\mathcal{L}_{\text{Align}} = -\frac{1}{\tau}\sum_{k=1}^{C} S_{ik}\,(z_i^\top \mu_k) \approx -\frac{1}{\tau}\, z_i^\top m_{\text{avg}},$$

$$m_{\text{avg}} = \frac{1}{C}\sum_{k=1}^{C} \mu_k.$$

Small proof (why $\|m_{\text{avg}}\| \ll 1$). Assume $\|\mu_k\| = 1$ and $\mu_j^\top \mu_k \approx 0$ for $j \neq k$. Then

$$\|m_{\text{avg}}\|^2 = \frac{1}{C^2}\Big(\sum_k \|\mu_k\|^2 + \sum_{j\neq k} \mu_j^\top \mu_k\Big) \approx \frac{1}{C^2}\cdot C = \frac{1}{C},$$

so $\|m_{\text{avg}}\| \approx 1/\sqrt{C} \ll 1$ for large C. Hence $\nabla_{z_i}\mathcal{L}_{\text{Align}} = -\frac{1}{\tau} m_{\text{avg}}$ is small.

3. Experimental Setup

3.1. Semantic Shift Benchmark

We used three widely recognized fine-grained image classification datasets to evaluate our method. These datasets are known for their subtle inter-class differences, which pose a significant challenge for machine learning models. To address this, it is essential to disentangle covariate shifts from semantic shifts as shown in Fig 3 [17].

Figure 3. This figure illustrates the distinction between semantic and covariate shifts. Motorcycles and Cars represent distinct semantic categories; their inherent meaning (category) remains unchanged even when external input conditions, such as lighting, undergo a covariate shift.

The core challenge of OSR lies in a model's ability to identify the "axes of variation" that distinguish between categories. For example, in a bird classification task, the model must learn which features define a species (e.g., beak shape, feather patterns) and which features are simply variations within a species (e.g., pose, lighting).

Many existing OSR benchmarks use datasets with abstract or broadly defined classes, making the "axes of variation" ambiguous. This can lead models to rely on semantically meaningless features, such as background color, to differentiate classes. In contrast, our proposed fine-grained setting provides a more constrained and semantically meaningful context. For example, in the CUB dataset, the classification axes are clearly defined by the unique features of bird species, making the problem better-posed and the evaluation more reliable.

3.2. Legacy Benchmarks

We follow the evaluation protocol from [14, 15, 23]. It has four datasets; each dataset is partitioned into five predefined splits, each split assigning a distinct set of N_{train} known classes. The remaining N_{test} classes in the dataset, excluding the selected known classes, serve as unknown classes for their corresponding split. To ensure robustness, we report the mean performance across these five splits.

Considering the complexity of real-world scenarios, where the ratio of seen to unseen data varies between tasks, we use openness to quantify the dataset's complexity[21].

Openness is defined as $1 - \sqrt{\frac{N_{train}}{N_{test}}}$. [14, 21]

We use the following datasets and splits: **MNIST/CIFAR10(C10)** [9, 11]: These datasets each contain 10 classes, with CIFAR10 featuring images of vehicles and animals. We randomly select six classes as known and treat the remaining four classes as unknown. The openness for this setup is 22.54%. **CIFAR+10(C+10)** [14] and **CIFAR+50(C+50)**: To increase the openness, we choose four classes from CIFAR10 as known and sample 10 and 50 additional classes from CIFAR100 as unknown, resulting in CIFAR+10 and CIFAR+50, with openness values of 46.55% and 72.78%, respectively. **Tiny-ImageNet(TIN)** [10]: Tiny-ImageNet, a subset of ImageNet, includes 200 classes. We select 20 classes as known, and the remaining 180 classes serve as unknown, leading to an openness of 68.37%.

3.3. Training Configuration

Similiar to [17], we used a cosine warm restarts scheduler with a warmup period and a total of 2 restarts over 600 epochs. A fixed random seed was used for reproducibility. SGD optimizer was used. The temperature was set to 0.07 as the default value based off previous implementations. It is summarized in Table 1.

3.3.1. Stage 01

SSB For ResNet, the feature dimension was 2048, however the projection dimension was 1024 consistently for all

datasets. The projector was a linear layer.

Legacy For CNN-32, the feature dimension was 128 for datasets, however the projection dimension was 128 for TinyImagenet, and 64 otherwise. The projector was a linear MLP layer.

3.3.2. Stage 02

SSB We followed the same training process as before, eith some minor exceptions: the learning rate being 0.01. Trained for 20 epochs. No Mixup/LS was applied. The classifier was a linear layer.

Legacy We followed the same training process as before, eith some minor exceptions: the learning rate being 0.01. Trained for 30 epochs. No Mixup/LS was applied. The classifier was a two layered MLP layer: with relu activation and the hidden dimension being the same as the input feature dimension. TinyImageNet uses leakyRelu, other datasets use Relu as in previous works.

3.4. Hyperparameter and Data Augmentation

We experimented with different loss functions and hyperparameters tailored to each dataset. A crucial component of our training was the use of **RandAugment**, a data augmentation technique that automatically finds the optimal combination of augmentation policies. The specific hyperparameters used for each loss function and dataset combination are detailed in Table 1. The RandAugment policy is controlled by two main parameters: N, the number of augmentation transformations to apply, and M, the magnitude of the transformations.

The data augmentation pipeline for training, which includes RandAugment, was as follows:

1. **Resize**: The image is resized to the target image size (e.g., 448x448).
2. **RandAugment**: A set of random augmentations (controlled by N and M) is applied to the image.
3. **RandomCrop**: A random crop of the image with padding is taken.
4. **RandomHorizontalFlip**: The image is randomly flipped horizontally.
5. **Normalize**: The image is normalized using a predefined mean and standard deviation.

The standard mean and standard deviation values used for normalization were (0.4914, 0.4822, 0.4465) and (0.2023, 0.1994, 0.2010), respectively. The test-time transformations were simpler, consisting of only resizing, converting to a tensor, and normalization.

4. Results

4.1. SSB

Extended SSB results with scoring rules KL[4], MSP and Energy[12] are provided in Table 2.

4.2. Legacy Benchmarks

Extended Legacy A results with all the scoring rules are provided in Table 3. Legacy B with the metrics AUIN/AUOUT presented in Table 4. For detailed explanation of the metrics, refer to [2].

5. Discussion

5.1. Normalized vs Unnormalized feature norms

To enable the classifier to better exploit norm-based separability to distinguish unknown samples, we use unnormalized feature norms during the training phase. This approach allows the classifier to adapt more effectively to the underlying structure of the data. We evaluate the impact of incorporating feature norms by comparing classifier performance with and without normalization. The results clearly indicate that using unnormalized features consistently improves OSCR and AUROC performance across all datasets. The Results are in Table 5.

5.2. Extended Ablation of Label Smoothing and Mixup

The extended ablation of Mixup and label smoothing for both pretrained and non-trained versions are as follows Table 6.

5.3. Visualizations and Extended AS/NS Analysis

To break down, we analyze the effects using our metrics AS and NS, the extended results for both pretrained and non-pretrained are in Table 7.

To visualize how label smoothing changes the feature norms as shown as density plots in Fig 4. In this, both of it leads to better separation between the unknown and known categories.

5.4. Extended Small-Batch Robustness Study

As observed, SpHOR maintains robustness and good performance when trained with small batches compared to SupCON8. Hard+Easy combined OSCR is used as the metric.

References

[1] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tao Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *Computer Vision – ECCV 2020*, pages 507–522. Springer, 2020. 6

[2] Guangyao Chen, Pai Peng, Xingyu Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 4, 5, 6

Table 1. Hyper-parameters for the benchmarks. Stage 01 We take the exact parameters from MLS[17] LR (Pre) is the learning rate for the pretrained models, LR is the learning rate for the non-pretrained models. The LR for MLS was increased to 0.01 for the non-pretrained models.

Loss	Dataset	Image Size	LR	RandAug N	RandAug M	Label Smoothing	Batch Size
MLS	CUB	448	0.001	2	30	0.3	64
MLS	FGV-Aircraft	448	0.001	2	15	0.2	64
MLS	SCARS	448	0.001/0.01	2	15	0.2	64
Sphor/SupCON	CUB	448	0.1	2	30	0.3	64
Sphor/SupCON	FGV-Aircraft	448	0.1	2	15	0.2	64
Sphor/SupCON	SCARS	448	0.1	2	15	0.2	64
ARPLoss	CUB	448	0.001	2	30	0.2	64
ARPLoss	FGV-Aircraft	448	0.001	2	18	0.1	64
ARPLoss	SCARS	448	0.001	2	18	0.1	64
Sphor	SVHN	32	0.1	1	18	0	128
Sphor	CIFAR-10	32	0.1	1	6	0	128
Sphor	CIFAR-10+/50+	32	0.1	1	6	0	128
Sphor	Tiny-Imagenet	64	0.1	1	9	0.9	128

Table 2. OSR results on the **Semantic Shift Benchmark**, on the ResNet50 backbone with/without Imagenet-pretraining. Each method (except ARPL+) is paired with a scoring rule (MaxLogit/KNN/PostMax/NNGuide) for unknown detection. The table reports *Closed-set Accuracy*, *AUROC*, and *OSCR*. Results are presented for both 'Easy/Hard' unknown test-data splits. The results were average over 3 different seeds. (w/o R_{Ortho}) denotes training without the orthogonality regularization term

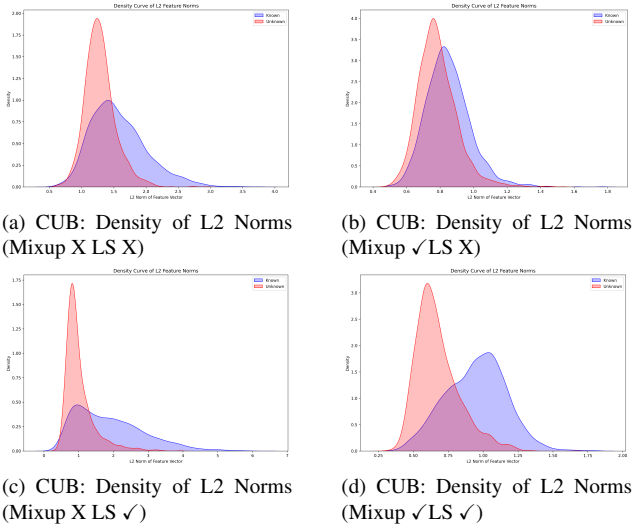
Semantic Shift	**Caltech-UCSD-Birds (CUB)[18]**			**Stanford Cars (SCars)[8]**			**FGVC-Aircraft[13]**		
Benchmark	Acc. ↑	AUROC ↑	OSCR ↑	Acc. ↑	AUROC ↑	OSCR ↑	Acc. ↑	AUROC ↑	OSCR ↑
With Imagenet Pretraining		(Easy/Hard)	(Easy/Hard)		(Easy/Hard)	(Easy/Hard)		(Easy/Hard)	(Easy/Hard)
ARPL+ [2]	85.4	81.77 / 73.88	73.07 / 66.94	89.8	84.95 / 76.39	79.32 / 72.21	83.3	85.75 / 74.57	75.51 / 66.69
MLS [17] +MaxLogit	85.2	83.16 / 73.98	74.38 / 67.16	89.3	83.96 / 76.63	78.32 / 72.33	80.2	85.23 / 73.73	73.01 / 73.01
MLS [17] + Mixup +MaxLogit	88.3	86.19 / 78.00	78.64 / 72.11	91.5	87.26 / 82.35	81.93 / 78.06	81.3	87.34 / 75.34	75.02 / 74.34
MLS [17] + Mixup +KL	88.3	85.32 / 74.45	77.14 / 68.16	91.5	84.4 / 79.47	79.15 / 75.08	81.30	86.12 / 74.43	73.14 / 72.44
MLS [17] + Mixup +Energy	88.3	85.32 / 74.45	77.11 / 68.15	91.5	84.45 / 79.37	79.19 / 75.01	81.3	86.23 / 73.99	73.12 / 72.32
MLS [17] + Mixup +MSP	88.3	85.82 / 79.21	78.65 / 73.55	91.5	88.16 / 82.66	83.49 / 79.07	81.3	87.21 / 75.01	74.12 / 73.22
MLS [17] + Mixup +PostMax	88.3	84.34 / 74.44	77.03 / 68.82	91.5	85.75 / 81.32	81.25 / 77.75	81.3	81.24 / 71.14	71.12 / 71.22
MLS [17] + Mixup +NNGuide	88.3	86.56 / 77.14	77.88 / 70.30	91.5	76.35 / 76.66	71.87 / 72.25	81.3	79.34 / 78.45	69.32 / 68.78
MLS [17] + Mixup +KNN	88.3	81.43 / 69.62	73.33 / 63.33	91.5	72.84 / 73.24	68.59 / 68.93	81.3	78.99 / 77.89	67.31 / 66.11
SupCON [6] +MaxLogit	78.2	88.52 / 75.44	72.62 / 63.55	91.8	90.90 / 79.25	85.61 / 75.42	88.9	83.02 / 79.93	77.79 / 75.36
SupCON [6] +PostMax	78.2	84.09 / 68.32	67.91 / 56.58	91.8	82.53 / 60.17	77.55 / 56.94	88.9	64.84 / 58.95	60.67 / 55.47
SupCON [6] +NNGuide	78.2	80.73 / 67.25	66.67 / 56.43	91.8	86.81 / 77.67	81.46 / 73.43	88.9	56.25 / 65.58	52.74 / 61.30
SupCON [6] +KNN	78.2	88.61 / 75.32	72.77 / 63.09	91.8	92.07 / 81.17	86.69 / 77.13	88.9	89.90 / 81.39	83.71 / 76.29
SupCON [6] +KL	78.2	88.84 / 73.83	72.53 / 61.76	91.8	90.64 / 78.83	85.26 / 74.85	88.9	89.83 / 80.38	83.68 / 75.54
SupCON [6] +Energy	78.2	88.82 / 73.83	72.53 / 61.76	91.8	90.64 / 78.82	85.26 / 74.85	88.9	75.79 / 76.54	71.11 / 71.92
SupCON [6] +MSP	78.2	88.4 / 76.17	72.57 / 64.32	91.8	90.94 / 79.25	85.66 / 75.46	88.9	90.62 / 80.72	84.54 / 76.25
SpHOR(w/o R_{Ortho}) +MaxLogit	90.3	91.10 / 83.59	85.09 / 79.10	96.6	94.62 / 83.31	92.51 / 82.00	89.4	90.62 / 82.26	85.00 / 78.07
SpHOR +MaxLogit	90.8	91.72 / 83.28	85.72 / 79.01	96.3	94.12 / 83.13	91.76 / 81.72	90.6	91.54 / 81.08	86.79 / 77.64
SpHOR +KNN	90.8	90.01 / 83.02	84.28 / 78.75	96.3	93.64 / 82.67	91.44 / 81.27	90.6	91.60 / 81.37	85.75 / 77.00
SpHOR +NNGuide	90.8	89.02 / 82.23	83.14 / 77.72	96.3	90.91 / 80.35	88.65 / 78.82	90.6	93.11 / 83.08	87.82 / 79.25
SpHOR +PostMax	90.8	91.35 / 81.83	85.35 / 77.60	96.3	93.44 / 82.57	91.12 / 81.17	90.6	93.02 / 81.74	87.56 / 77.94
SpHOR +KL	90.8	91.67 / 82.68	85.52 / 78.25	96.3	93.86 / 81.76	91.57 / 80.36	90.6	93.09 / 82.88	87.43 / 78.72
SpHOR +Energy	90.8	83.77 / 76.68	78.22 / 72.25	96.3	88.96 / 80.96	86.66 / 79.14	90.6	86.32 / 76.08	81.58 / 72.41
SpHOR +MSP	90.8	91.35 / 83.49	85.54 / 79.21	96.3	94.38 / 82.42	92.18 / 81.12	90.6	91.54 / 81.08	86.79 / 77.64
Without Imagenet Pretraining		(Easy/Hard)	(Easy/Hard)		(Easy/Hard)	(Easy/Hard)		(Easy/Hard)	(Easy/Hard)
ARPL+ [2]	46.8	67.59 / 59.35	37.33 / 33.95	76.1	80.45 / 72.38	66.70 / 61.18	77.8	78.53 / 70.18	67.33 / 61.41
MLS [17] +MaxLogit	53.2	68.42 / 61.79	43.44 / 41.26	76.5	79.31 / 71.22	65.12 / 60.34	78.2	78.66 / 72.23	67.45 / 62.89
MLS [17] + Mixup +MaxLogit	56.9	70.22 / 63.69	45.74 / 42.57	86.9	87.38 / 77.83	79.54 / 71.91	82.1	84.44 / 78.39	74.48 / 70.23
MLS [17] + Mixup +PostMax	56.9	73.10 / 62.93	47.10 / 42.01	86.9	86.39 / 77.13	78.76 / 71.32	82.1	80.65 / 73.86	72.30 / 67.28
MLS [17] + Mixup +NNGuide	56.9	59.38 / 59.65	38.75 / 38.66	86.9	81.37 / 74.81	74.10 / 68.56	82.1	77.49 / 79.10	67.45 / 68.34
MLS [17] + Mixup +KNN	56.9	52.53 / 53.96	31.01 / 31.80	86.9	63.42 / 65.01	57.88 / 59.16	82.1	79.17 / 78.92	67.81 / 67.55
MLS [17] + Mixup +MSP	56.9	70.69 / 65.39	46.80 / 44.47	86.9	87.34 / 78.17	79.76 / 72.51	82.1	84.00 / 78.38	74.50 / 70.67
MLS [17] + Mixup +Energy	56.9	67.87 / 60.62	42.93 / 39.13	86.9	85.86 / 76.16	77.36 / 69.53	82.1	84.88 / 78.17	73.80 / 68.98
MLS [17] + Mixup +KL	56.9	67.82 / 60.60	42.90 / 39.12	86.9	85.84 / 76.15	77.35 / 69.51	82.1	84.85 / 78.24	73.79 / 69.02
SupCON [6] +MaxLogit	77.8	87.86 / 77.06	71.73 / 64.71	78.7	75.17 / 65.74	63.73 / 56.42	66.7	82.55 / 77.71	59.64 / 57.04
SupCON [6] +PostMax	77.8	85.02 / 69.59	68.75 / 57.87	78.7	75.83 / 68.97	62.78 / 57.93	66.7	45.95 / 66.78	31.61 / 46.27
SupCON [6] +KNN	77.8	88.22 / 76.89	72.25 / 64.40	78.7	81.86 / 71.14	68.68 / 60.71	66.7	84.39 / 78.89	60.84 / 58.10
SupCON [6] +NNGuide	77.8	81.24 / 70.64	66.63 / 58.94	78.7	71.23 / 63.56	60.08 / 54.02	66.7	68.79 / 71.57	50.41 / 52.62
SupCON [6] +MSP	77.8	87.75 / 77.53	71.7 / 65.21	78.7	78.83 / 70.63	67.25 / 61.46	66.7	81.79 / 77.80	59.23 / 57.22
SupCON [6] +Energy	77.8	88.32 / 75.37	71.66 / 62.88	78.7	73.13 / 63.98	61.85 / 54.62	66.7	84.54 / 76.97	60.40 / 56.14
SupCON [6] +KL	77.8	88.33 / 75.36	71.66 / 62.87	78.7	73.14 / 63.99	61.85 / 54.63	66.7	84.52 / 76.97	60.39 / 56.14
SpHOR(w/o R_{Ortho}) +MaxLogit	80.5	86.93 / 75.19	74.49 / 66.59	94.6	92.77 / 80.78	89.52 / 78.77	88.8	89.72 / 82.55	83.76 / 78.03
SpHOR +MaxLogit	82.7	87.47 / 77.52	76.72 / 70.02	94.4	93.01 / 81.08	89.60 / 78.97	88.3	89.67 / 81.46	83.41 / 76.93
SpHOR +MSP	82.5	84.12 / 76.35	73.70 / 68.3	94.4	92.21 / 80.32	89.32 / 78.52	88.3	89.03 / 81.73	82.97 / 77.25
SpHOR +Energy	82.7	75.93 / 67.57	64.27 / 57.59	94.4	84.30 / 77.90	80.89 / 74.85	88.3	88.28 / 80.25	81.25 / 75.27
SpHOR +KL	82.7	83.73 / 75.89	74.69 / 69.03	94.4	91.73 / 79.92	88.72 / 78.09	88.3	89.32 / 81.54	82.94 / 76.73
SpHOR +PostMax	82.7	87.30 / 76.76	76.54 / 69.19	94.4	92.78 / 80.59	89.43 / 78.52	88.3	89.89 / 79.25	83.64 / 74.86
SpHOR +NNGuide	82.7	83.33 / 77.82	72.82 / 68.94	94.4	91.54 / 80.84	88.41 / 78.75	88.3	86.33 / 80.28	81.15 / 76.15
SpHOR +KNN	82.7	84.96 / 77.77	75.38 / 70.29	94.4	91.80 / 80.94	88.89 / 79.12	88.3	88.98 / 80.55	82.67 / 76.09

Table 3. Benchmark A with different scoring rules.

Method	SVHN	CIF10	CIF+10	CIF+50	TIN
Openness%	*22.5*	*22.5*	*46.5*	*72.7*	*68.3*
SpHOR +Energy	98.6	92.7	98.3	97.3	84.1
SpHOR +NNguide	97.2	94.7	98.2	97.3	83.0
SpHOR +KL	98.7	94.9	98.3	97.3	83.7
SpHOR +KNN5	99.0	94.5	98.2	97.0	81.6
SpHOR +MSP	98.7	93.7	97.4	96.4	82.1
SpHOR +PostMax	93.1	85.5	91.3	88.8	79.3

Method	DTACC ↑	AUROC ↑	AUIN ↑	AUOUT ↑
Easy	Known:	CIFAR10	Unknown:	SVHN
SoftMax[3]	86.4	90.6	88.3	93.6
GCPL[22]	86.1	91.3	86.6	94.8
RPL[1]	87.1	92.0	89.6	95.1
ARPL[2]	91.6	96.6	94.8	98.0
CSI[16]	92.8	97.9	96.2	99.0
OpenGAN[7]	92.1	95.9	93.4	97.1
CSSR[5]	95.7	99.1	98.2	**99.6**
RCSSR[5]	95.7	99.1	98.3	**99.6**
MEDAF[20]	95.3	99.1	98.0	**99.6**
BackMix[20]	88.5	94.1	93.5	97.5
SpHOR	97.5	99.1	99.4	97.7
SpHOR (+R_{Ortho})	**97.6**	**99.6**	**99.7**	99.2
Hard	Known:	CIFAR10	Unknown:	CIFAR100
SoftMax[3]	79.8	86.3	88.4	82.5
GCPL[22]	80.2	86.4	86.6	84.1
RPL[1]	80.6	87.1	88.8	83.8
ARPL[2]	83.4	90.3	91.1	88.4
CSI[16]	84.4	91.6	92.5	90.0
OpenGAN[7]	84.2	89.7	87.7	89.6
CSSR[5]	83.8	92.1	89.4	89.3
RCSSR[5]	85.3	92.3	92.9	91.0
MEDAF[20]	85.4	92.5	93.2	91.1
BackMix[20]	84.9	91.3	93.0	88.1
SpHOR	**86.7**	92.8	93.0	91.1
SpHOR (+R_{Ortho})	86.4	**93.2**	**93.9**	**91.5**

Table 4. Evaluation is conducted on the Legacy Benchmark B [2], designed for OSR across different validation configurations. Specifically, this includes scenarios where CIFAR-10 constitutes known classes, and CIFAR-100 or SVHN represent unknown classes. The results, employing a CNN-32 backbone[20]



(a) CUB: Density of L2 Norms (Mixup X LS X)

(b) CUB: Density of L2 Norms (Mixup ✓LS X)

(c) CUB: Density of L2 Norms (Mixup X LS ✓)

(d) CUB: Density of L2 Norms (Mixup ✓LS ✓)

Figure 4. Distribution of Feature L2 norms under different combinations of Mixup and Label Smoothing on the CUB dataset. Each plot shows how regularization strategies affect the magnitude of learned representations.

[3] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of the International Conference on Learning Representations*, pages 1–12, 2016. 6

[4] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8759–8773. PMLR, 2022. 4

[5] Hongzhi Huang, Yu Wang, Qinghua Hu, and Ming-Ming Cheng. Class-specific semantic reconstruction for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 6

[6] Prannay Khosla et al. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, pages 18661–18673, 2020. 5

[7] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 813–822, 2021. 6

[8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 554–561. IEEE, 2013. 5

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 3

[10] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. https://tiny-imagenet.herokuapp.com/, 2015. 3

[11] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 3

[12] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 21464–21475, 2020. 4

[13] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013. 5

[14] Lawrence Neal, Matthew L. Olson, Xiaoli Z. Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision*, pages 620–635, 2018. 3

[15] Pramuditha Perera, Vlad I. Morariu, Rameswar Jain, Varun Manjunatha, Curtis Wigington, Vicente Ordonez, and Vishal M. Patel. Generative-discriminative feature representations for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11814–11823, 2020. 3

[16] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo

	CUB			SCars			FGVC-Aircraft		
	Acc.	AUROC	OSCR	Acc.	AUROC	OSCR	Acc.	AUROC	OSCR
$z/\|z\|_2$	**91.1**	90.13 / 80.12	84.5 / 76.21	**96.5**	92.58 / 80.99	90.6 / 79.72	**90.6**	91.53 / 81.08	86.78 / 77.64
z	90.8	**91.72 / 83.28**	**85.72 / 79.01**	96.3	**94.12 / 83.13**	**91.76 / 81.72**	**90.6**	**91.54 / 81.08**	**86.79 / 77.64**

Table 5. Classifier performance on CUB, SCars, and FGVC-Aircraft using normalized vs. unnormalized features. OSR/AUROC shown as Easy/Hard. The performance for Aircraft is what similar, although accuracy improves with normalized features. This suggests that the L2 norm carries valuable information about the model's confidence.

	CUB			SCars			FGVC-Aircraft		
	Acc.	AUROC	OSCR	Acc.	AUROC	OSCR	Acc.	AUROC	OSCR
Pretrained									
SpHOR(No Mix/LS)	84.1	84.86 / 77.61	75.8 / 70.4	94.9	88.58 / 78.94	86.57 / 77.51	89.70	87.38 / 81.72	82.80 / 77.87
SpHOR(LS)	83.4	87.63 / 77.00	77.59 / 69.91	94.2	92.44 / 78.05	89.46 / 76.26	89.4	90.61 / 82.26	85.00 / 78.06
SpHOR(Mix)	88.1	88.3 / 78.36	80.71 / 72.85	96.3	94.12 / 83.13	91.75 / 81.71	90.6	91.53 / 81.08	86.78 / 77.64
SpHOR (Mix/LS)	**91.2**	**91.1 / 83.77**	**85.41 / 79.65**	**96.7**	**94.47 / 83.48**	**92.36 / 82.13**	**90.6**	**91.54 / 81.08**	**86.79 / 77.64**
Not pretrained									
SpHOR(No Mix/LS)	79.7	79.76 / 72.12	69.72 / 64.02	93.3	88.03 / 78.1	85.22 / 76.2	85.6	83.54 / 76.79	76.42 / 71.24
SpHOR(LS)	83.4	87.63 / 77.00	77.59 / 69.91	92.7	90.35 / 79.05	86.91 / 76.83	82.6	83.91 / 76.45	74.25 / 68.73
SpHOR(Mix)	78.2	82.7 / 74.28	70.97 / 65.2	94.7	91.25 / 81.96	88.27 / 79.91	84.7	87.6 / 79.72	79.36 / 73.67
SpHOR (Mix/LS)	**83.4**	**86.01 / 77.16**	**76.48 / 70.12**	**94.0**	**91.35 / 79.78**	**88.31 / 77.77**	**88.0**	**90.25 / 82.69**	**83.55 / 77.58**

Table 6. Ablation study on the effects of Label Smoothing (LS) and Mixup when applied to representation learning within the SpHOR method (With MaxLogit Scoring rule). Results are averaged across three datasets from the SSB benchmark (SCARS, Aircraft, and CUB). Closed-set Accuracy, AUROC, and OSCR are reported for both easy and hard splits.

Table 7. Ablation study reporting *Angular Separability (AS)* and *Norm Separability (NS)* across SSB datasets under different **Mixup and Label Smoothing (LS)** settings. For both pretrained and non-pretrained cases.

Mixup	LS	**CUB**		**SCARS**		**Aircraft**	
		AS ↓	NS ↑	AS ↓	NS ↑	AS ↓	NS ↑
Raw							
x	x	0.7117	66.13	0.7705	77.88	0.7730	71.54
✓	x	0.6607	55.94	0.7031	61.99	0.7211	56.36
x	✓	0.7533	74.79	0.7715	85.91	0.8003	81.96
✓	✓	0.7182	77.47	0.7621	86.62	0.7705	83.84
Pretrained							
x	x	0.6869	71.30	0.7765	84.18	0.7504	71.06
✓	x	0.6340	65.37	0.6917	72.48	0.6717	55.80
x	✓	0.7457	79.14	0.7824	86.92	0.8044	82.92
✓	✓	0.7121	78.67	0.7268	90.43	0.7197	84.96

Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 11839–11852, 2020. 6

[17] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *Proceedings of the International Conference on Learning Representations*, pages 1–14, 2022. 3, 5

[18] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011. 5

[19] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020. 1

[20] Yu Wang, Junxian Mu, Pengfei Zhu, and Qinghua Hu. Exploring diverse representations for open set recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5731–5739, 2024. 6

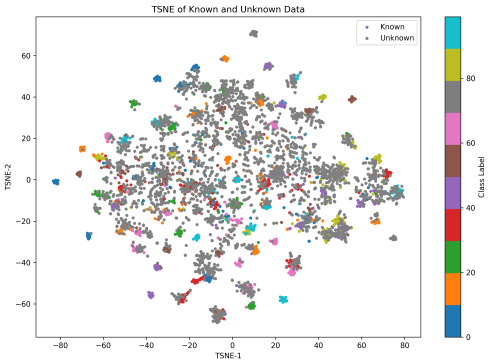
[21] Baile Xu, Furao Shen, and Jian Zhao. Contrastive open set recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10546–10556, 2023. 3

[22] Hong-Ming Yang, Xiao-Yan Zhang, Fei Yin, Qing Yang, and Cheng-Lin Liu. Convolutional prototype network for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2358–2370, 2020. 6

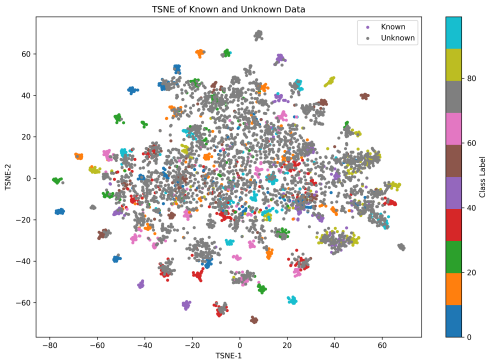
[23] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4016–4025, 2019. 3

Batch Size	64	32	16
	SpHOR/SupCON	SpHOR/SupCON	SpHOR/SupCON
CUB	78.57 / 75.11	76.81 / 76.72	77.10 / 62.38
AIR	78.71 / 72.42	81.43 / 71.40	81.80 / 62.90
SCARS	75.97 / 84.55	86.05 / 84.63	88.27/ 81.92

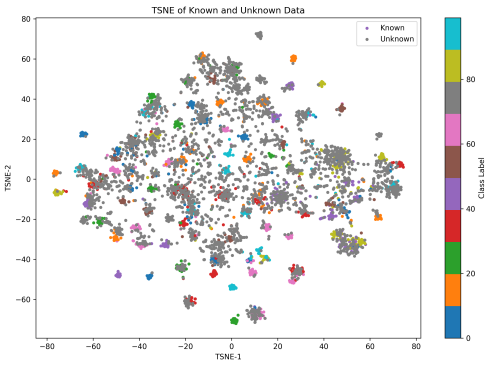
Table 8. Batch size performance comparison with OSCR for SSB. Trained for 200 epochs.



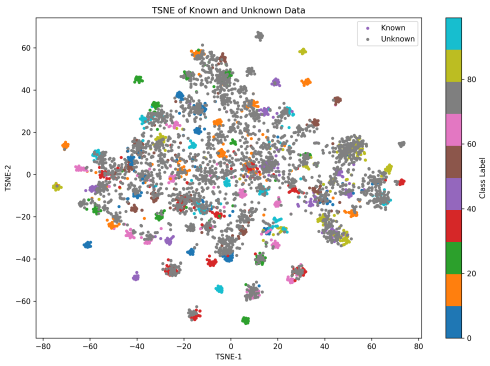
(a) Normalized (Mixup X LS X)



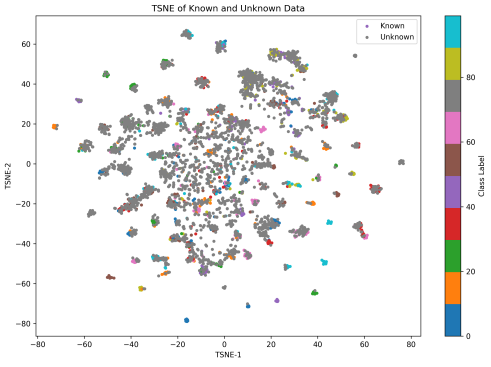
(b) Raw (Mixup X LS X)



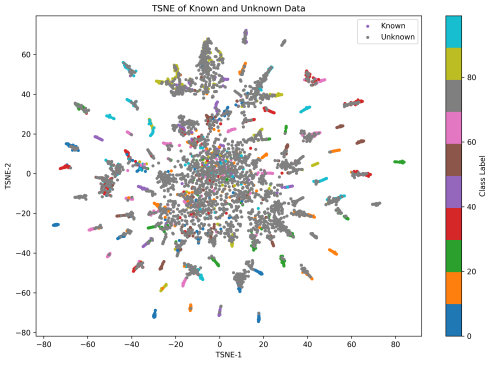
(c) Normalized (Mixup ✓LS X)



(d) Raw (Mixup ✓LS X)

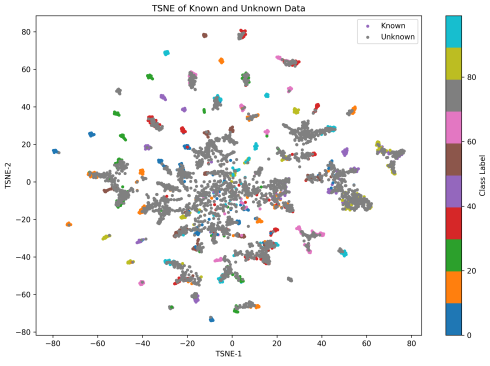


(e) Normalized (Mixup X LS ✓)



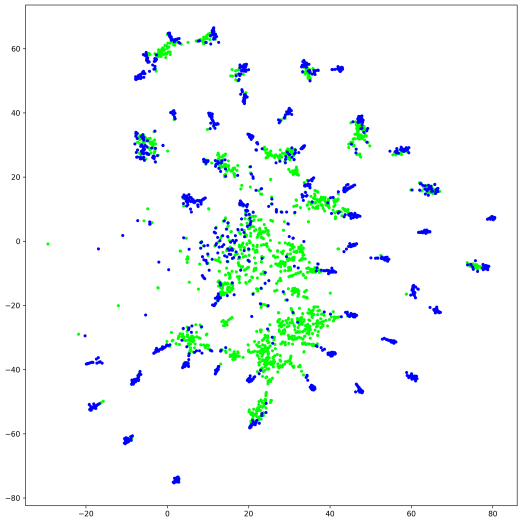
(f) Raw (Mixup X LS ✓)

(g) Normalized (Mixup ✓LS ✓)

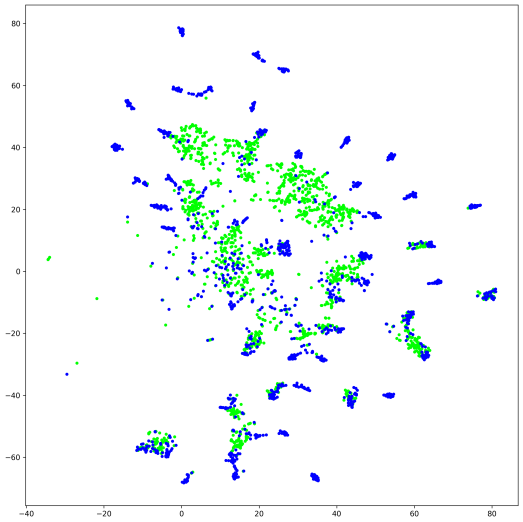


(h) Raw (Mixup ✓LS ✓)

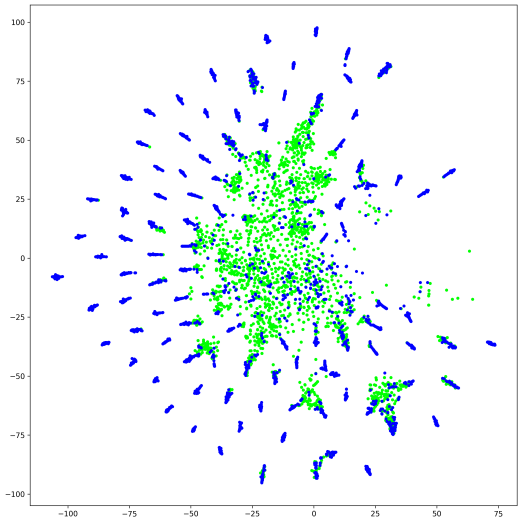
Figure 5. TSNE visualizations of normalized and raw features for different combinations of Mixup and Label Smoothing on the CUB dataset.



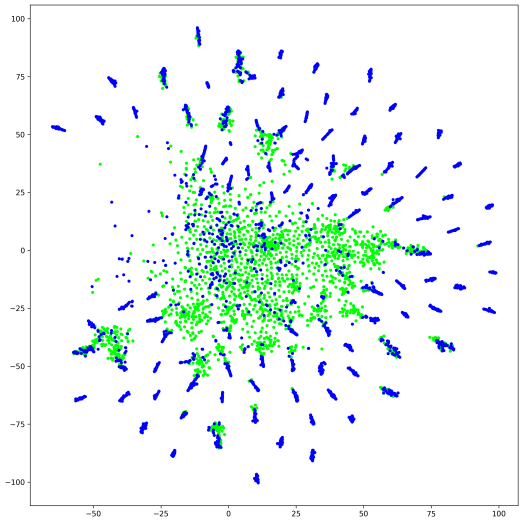
(a) Aircraft (w/o R_{Ortho})



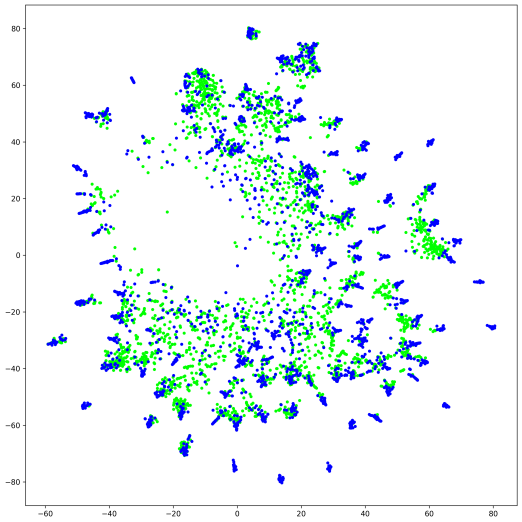
(b) Aircraft



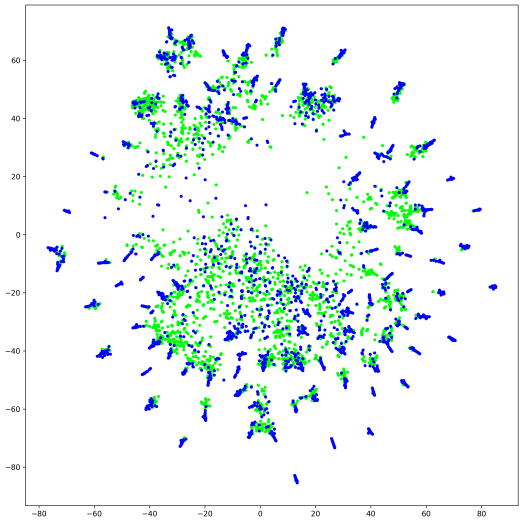
(c) Scars (w/o R_{Ortho})



(d) Scars



(e) CUB (w/o R_{Ortho})



(f) CUB

Figure 6. TSNE visualizations of features with and without orthogonality (w/o R_{Ortho})