# Inducing Early Neural Collapse in Deep Neural Networks for Improved Out-of-Distribution Detection

**Anonymous for double-blind review**


## Abstract

We propose a simple modification to standard ResNet architectures–L2 regularization over feature space–that substantially improves out-of-distribution (OoD) performance on the previously proposed Deep Deterministic Uncertainty (DDU) benchmark. We show that this change also induces early Neural Collapse (NC), an effect under which better OoD performance is more probable. Our method achieves comparable or superior OoD detection scores and classification accuracy in a small fraction of the training time of the benchmark. Additionally, it substantially improves worst case OoD performance over multiple, randomly initialized models. Though we do not suggest that NC is the sole mechanism or a comprehensive explanation for OoD behaviour in deep neural networks (DNN), we believe NC's simple mathematical and geometric structure can provide a framework for analysis of this complex phenomenon in future work.


## 1 Introduction

It is well known that Deep Neural Networks (DNNs) lack robustness to distribution shift and may not reliably indicate failure when receiving out of distribution (OoD) inputs Rabanser et al. (2018), Chen et al. (2020). Specifically, networks may give confident predictions in cases where inputs are completely irrelevant, e.g. an image of a plane input into a network trained to classify dogs or cats maybe produce high confidence scores for either dogs or cats. This inability for networks to know what they do not know hinders the application of machine learning in engineering and other safety critical domains Henne et al. (2020).

A number of recent developments have attempted to address this problem, the most widely used being Monte Carlo Dropout (MCD) and ensembles Gal and Ghahramani (2016), Lakshminarayanan et al. (2017). While supported by a reasonable theoretical background, MCD lacks performance in some applications and requires multiple forward passes of the model after training Haas and Rabus (2021), Ovadia et al. (2019). Ensembles can provide better accuracy than MCD, as well as better OoD detection under larger distribution shifts, but require a substantial increase in compute Ovadia et al. (2019).

These limitations have spurred interest in deterministic and single forward pass methods. Notable amongst these is Deep Deterministic Uncertainty (DDU) Mukhoti et al. (2021). DDU is much simpler than many competing approaches, Liu et al. (2020), Van Amersfoort et al. (2020), van Amersfoort et al. (2021), produces competitive or state-of-the-art results and has been proposed as a benchmark for uncertainty methods. A limitation, as shown in our experiments, is that DDU requires long training times and produces models with inconsistent performance.

We demonstrate that DDU can be substantially improved via L2 regularization over feature space in standard ResNet architectures. Beyond offering performance gains in accuracy and OoD detection, L2 regularization induces neural collapse (NC) much earlier than standard training. NC was recently found to occur in many NN architectures when they are overtrained Papyan et al. (2020). This may provide a way to render the complexity of deep neural networks more tractable, such that they can be analyzed through the relative geometric and mathematical simplicity of simplex Equiangular Tight Frames (simplex ETF) Mixon et al. (2022), Zhu et al. (2021), Lu and Steinerberger (2020), Ji et al. (2021). Although this simplex ETF is limited to the feature layer and decision classifier, these layers summarize a substantial amount of network functionality. While Papyan et al. demonstrate increased adversarial robustness under NC, to the

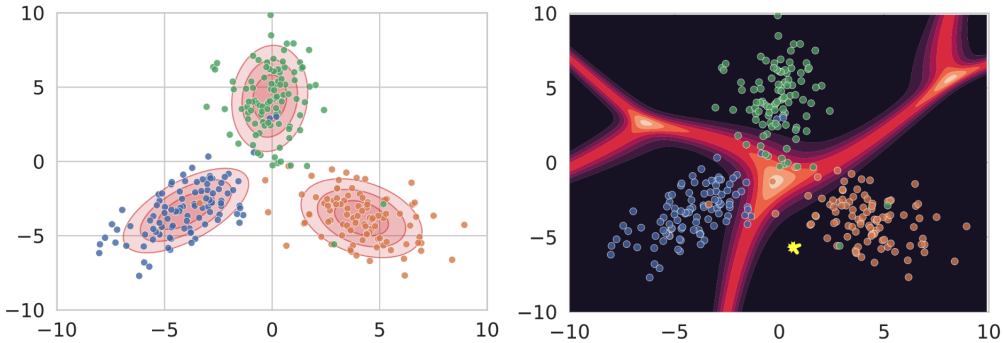


Figure 1: An illustration of the DDU method from Mukhoti et al. (2021) Left: In this hypothetical example with a two-dimensional feature space, DDU fits Gaussians over each of three classes as the components of a GMM, $q(y, z)$. Right: with standard decision boundaries (red), embeddings in this space that are far from class centroids (yellow points) are labelled with high confidence (darker is higher confidence).

best of our knowledge, we present the first study of the relationship between OoD detection and NC.

We summarize our contributions as follows:

1) L2 regularization over the feature space of deep learning models results in OoD detection and classification performance that is competitive with or exceeds performance of the DDU benchmark. Most notably, the lower bound on OoD performance across model seeds is substantially improved in all cases.

2) Models trained with L2 regularization over feature space produce the aforementioned performance results in 17% (ResNet18) to 29% (ResNet50) of the training time of the DDU benchmark.

3) L2 regularization over feature space induces NC as much as five times faster than standard training. Controlling the rate of NC may be useful for analyzing DNN behaviour.

4) NC encourages OoD detection under our proposed modification to the DDU method. We show evidence that fast NC plays a role in achieving OoD detection performance with less training, and that training directly on NC has a substantially different effect on OoD performance than standad cross entropy (CE) training. This connection between simplex ETFs that naturally arise in DNNs and OoD performance permits an elegant analytical framework for further study of the underlying mechanisms that govern uncertainty and robustness in DNNs.

## 2 Background

### 2.1 Problem Definition

A standard classification model maps images to classes $f : x \to y$, such that $x \in X$ where $X$ is the set of images and $y \in \{1, ..., k\}$ where there are $k$ possible classes. The model is composed of a feature extractor which embeds images into feature space $z$ and a linear decision classifier that transforms the feature space into a vector of length $k$. These logits are run through a softmax function to produce a probability over classes over which the $argmax$ function identifies the class prediction $y$. We note that the set of images can be broken into $X \in X_{in}$ which are images drawn from the same distribution as the training data, and $X \in X_{out}$, which are all other images. For OoD detection, we can conceive of the the model as learning a data-generating distribution $p(y|x)$ that is composed of in and out of distribution components Liu et al. (2020):

$$p(y|x) = p(y, x \in X_{in}) + p(y, x \in X_{out}) \tag{1}$$

| | AUROC | | | | | | | | | Accuracy | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SVHN | | | CIFAR100 | | | Tiny ImageNet | | | CIFAR10 Test | | |
| **ResNet18** | **No L2 350** | **No L2 60** | **L2 60** | **No L2 350** | **No L2 60** | **L2 60** | **No L2 350** | **No L2 60** | **L2 60** | **No L2 350** | **No L2 60** | **L2 60** |
| **Min** | 0.877 | 0.848 | **0.924** | 0.861 | 0.796 | **0.878** | 0.881 | 0.809 | **0.898** | **0.928** | 0.881 | 0.924 |
| **Max** | 0.949 | 0.953 | **0.961** | 0.885 | 0.845 | **0.886** | 0.909 | 0.872 | **0.911** | **0.941** | 0.912 | 0.929 |
| **Mean** | 0.915±.018 | 0.912±.035 | **0.938**±.010 | 0.875±.008 | 0.824±.014 | **0.881**±.002 | 0.894±.009 | 0.841±.015 | **0.904**±.004 | **0.935**±.005 | 0.898±.008 | 0.927±.001 |
| **ResNet50** | **No L2 350** | **No L2 100** | **L2 100** | **No L2 350** | **No L2 100** | **L2 100** | **No L2 350** | **No L2 100** | **L2 100** | **No L2 350** | **No L2 100** | **L2 100** |
| **Min** | 0.903 | 0.869 | **0.927** | 0.817 | 0.794 | **0.892** | 0.881 | 0.852 | **0.912** | 0.930 | 0.896 | **0.937** |
| **Max** | **0.987** | 0.980 | 0.955 | 0.891 | 0.866 | **0.896** | 0.909 | 0.905 | **0.918** | **0.946** | 0.927 | 0.943 |
| **Mean** | **0.955**±.026 | 0.944±.029 | 0.945±.007 | 0.871±.018 | 0.837±.022 | **0.894**±.001 | 0.894±.020 | 0.880±.016 | **0.915**±.002 | 0.939±.006 | 0.916±.008 | **0.941**±.002 |

Table 1: OoD detection and classification accuracy results for ResNet18 and ResNet50 models, 15 seeds per experiment, trained on CIFAR10, and SVHN, CIFAR100 and Tiny ImageNet test sets used as OoD data. For all models, we indicate whether L2 regularization over feature space was used (L2/No L2) and how many training epochs occurred (60/100/350). We compare our method with DDU baseline models (No L2 350). Note that the variability of AUROC scores is substantially reduced under L2 regularization of feature space, in particular, lower bounds increase.

Under this regime, we desire that the model learns a density over feature space, where ID embeddings occupy class-wise high density areas, and OoD embeddings occupy low density areas far away from the ID embeddings under a meaningful distance metric. It is well known that standard deep neural networks produce arbitrarily high confidence scores for inputs that map to feature space far from the training data, which is obviously undesirable Hein et al. (2018).

To evaluate models, we merge ID and OoD images into a single test set. OoD performance is then considered a binary classification task, where we measure how well OoD images can be separated from ID images using a score derived from our model. As with the DDU benchmark, we use the log probabilities generated by the fitted GMM and evaluate using area under ROC curve using CIFAR10, CIFAR100, SVHN and Tiny ImageNet datasets.

### 2.2 Deep Deterministic Uncertainty

Deep Deterministic Uncertainty (DDU) was proposed by Mukhoti et al. as a simple but competitive method to detect OoD samples Mukhoti et al. (2021). It requires only a single pass through a single network. To evaluate uncertainty, it uses a class-wise Gaussian mixture model (GMM) $q(y, z)$, with mean and covariance parameters retrieved from the feature layer via a one-time pass over the training set from a converged model. At test time, feature densities $z$ are evaluated under the GMM as $q(z) = \sum_y q(z|y)q(y)$. This log probability used as the scoring method for an area under ROC curve to assess separability of ID and OoD data, where the OoD data is drawn from standard machine learning community datasets.

Van Amersfoort et al. hypothesize that *feature collapse*–the case wherein different images are embedded into the same location in feature space–is a significant reason that uncertainty and OoD estimation methods fail Van Amersfoort et al. (2020). They argue that feature collapse can be mitigated through the management of *sensitivity* and *smoothness*. The former refers to the network differentiating different images in feature space, and the latter to ensuring that small differences in images do not result in large differences in feature space position. Together, these can be thought of as a bi-lipschitz constraint over features with respect to inputs. More broadly, it is an attempt to impose a metric space over features. The motivation here is simply that, if achieved, high density regions of embedding space would contain ID images, low density regions would contain OoD images, and one could then measure the relevant distances in embedding space to determine whether an input sample was drawn from the training dataset.

DDU employs spectral normalization to enforce bi-lipschitzness, in addition to Leaky ReLUs and strided average pooling instead of 1x1 convolutions to perform downsampling in residual connections Liu et al. (2020). These latter interventions are introduced to increase sensitivity. In section IV we show that L2 regularization over feature space further improves the effect of the DDU regularization scheme, and that these improvements are connected with neural collapse.

| Intervention | | | AUROC | | | Accuracy | | |
|---|---|---|---|---|---|---|---|---|
| **Spectral Norm** | **Leaky/GAP** | **L2 Norm** | **Min** | **Max** | **Mean** | **Min** | **Max** | **Mean** |
| ✗ | ✗ | ✗ | 0.855 | 0.939 | 0.910 | 0.884 | 0.927 | 0.916 |
| | | ✓ | 0.894 | 0.951 | 0.932 | 0.924 | 0.930 | 0.927 |
| | ✓ | ✗ | 0.830 | **0.965** | 0.889 | 0.872 | 0.909 | 0.898 |
| | | ✓ | 0.917 | 0.958 | **0.944** | **0.925** | 0.929 | 0.927 |
| ✓ | ✗ | ✗ | 0.884 | 0.954 | 0.923 | 0.904 | 0.926 | 0.917 |
| | | ✓ | 0.903 | 0.959 | 0.934 | **0.925** | **0.931** | **0.928** |
| | ✓ | ✗ | 0.848 | 0.953 | 0.912 | 0.881 | 0.912 | 0.898 |
| | | ✓ | **0.924** | 0.961 | 0.938 | 0.924 | 0.929 | 0.927 |

Table 2: ResNet18 ablation study of effects from the DDU benchmark along with our method, 15 seeds per experiment. Models trained on CIFAR10, with SVHN as OoD data. When combined with any other interventions, our method improves lower bounds and average performance.

### 2.3 Neural Collapse

Papyan et al. (2020) recently observed a phenomenon known as Neural Collapse that occurs when networks are overtrained i.e. training continues well after cross-entropy loss goes to zero on the training set. NC has four properties:

NC1: Variability collapse: the intra-class covariance of each class in feature space approaches zero.

NC2: Convergence to a simplex equiangular tight frame (ETF): the angles between each class mean are maximized and equal and the distances of each class mean from the global mean of classes are equal, i.e. class means are placed at maximally equiangular locations on a hypersphere.

NC3: Convergence to self-duality: model decision regions and class means converge to a symmetry where each class mean occupies the center of it's decision region, and all decision regions are equally sized.

NC4: Simplification to Nearest Class Center (NCC): the classifier assigns the highest probability for a given point in feature space to the nearest class mean.

The measurements for each of these properties is outlined in Section III.

### 2.4 Related Work

Until recently, most research toward uncertainty estimation in deep learning took a Bayesian approach. The high number of parameters in DL models renders posterior integration intractable, so approximations (typically variational inference) have been used Gal and Ghahramani (2016). Monte Carlo Dropout (MCD), which draws a connection between test time dropout and variational inference, has emerged as a popular method to estimate uncertainty Gal and Ghahramani (2016). Despite MCD's simplicity, scalability and applicability to nearly any model architecture, it is often outperformed by deterministic ensembles Ovadia et al. (2019). Lakshminarayanan et al. observed that a simple average over the predictions of multiple deterministic models with the same architecture but unique parameter initialisations produces a competitive uncertainty estimate Lakshminarayanan et al. (2017).

The compute required for running multiple model passes at test time and for training and testing ensembles has motivated recent research around single model, single pass uncertainty estimates. Van Amersfoort et al. enforce a bi-lipschitz penalty on gradients during training to create a distance-aware feature space, which is then measured with a radial basis function (RBF) instead of a standard linear classifier Van Amersfoort et al. (2020). While showing some promise, bi-lipschitz penalties can be unstable during training Mukhoti et al. (2021). In later work, van Amersfoort et al. (2021) replace the RBF apparatus with a deep kernel and point Gaussian Process (GP), while still enforcing distance-awareness with spectral normalization. A

similar approach by Liu et al. (2020) uses a Random Fourier Feature approximation to construct the GP, but results are not as competitive.

Lee et al. (2018) propose fitting a class-wise conditional Gaussian with shared covariance matrix to multiple layers, but employ OoD and adversarial data to learn a weighted average of scores over layers. They also add noise to inputs at test time to enhance OoD separability.

Parseval tight frames have been proposed to induce a lipschitz constraint on feature layers Cisse et al. (2017). These were found to reduce training time and improve adversarial robustness, but require constraining each layer's weights to a Parseval tight frame.

L2 regularization and the notion of a hypersphere embedding space have been explored elsewhere. Liu et al. (2017) proposed using an angular margin softmax that enforced low intraclass and high interclass decision separation on a hypersphere for face recognition. They found that a distance-aware feature space arises from the induced hypersphere manifold and produced better results than (at the time) state-of-the-art methods that used explicit metric losses. We note that our method is much simpler in that it requires no hyperparameter tuning and no changes to the loss function. Wei et al. (2022) introduce a normalized cross-entropy loss function to decouple magnitude and direction in the logit (decision) layer, which produces competitive results for OoD detection using softmax scores or metrics that depend on them. Though it uses L2 normalization to do so, it remains substantially different from our method in that we do not modify the loss function and apply normalization only to the feature layer.

## 3 Methodology

### 3.1 Models and Training

For all baselines we used the ResNet18 and ResNet50 models provided in the DDU benchmark Mukhoti et al. (2021). All models (except those explicitly noted in the ablation study) use spectral normalization, leaky ReLUs and Global Average Pooling (GAP), as these produce the strongest baselines. Each experiment was conducted with fifteen randomly initialized model parameter sets; no fixed seeds were used at any time for initialization. We set the batch size to 1024 for all training runs, except the NC intervention models, which were more stable when training with a batch size of 2048. All training was conducted on four NVIDIA V100 GPUs in PyTorch 1.10.1 Paszke et al. (2019).

Stochastic gradient descent (SGD) with an initial learning rate of $1e^{-1}$ was used as the optimizer for all experiments. We used a learning rate schedule that decreased by one order of magnitude at 150 and 250 epochs for the 350 epoch models, as per the DDU benchmark. We adjust the learning rate at 75 and 90 for the 100 epoch ResNet50 models, and at 40 and 50 for the 60 epoch ResNet18 models. Models were trained on the standard CIFAR-10 training data set with a validation size of 10% created with a fixed random seed.

All baselines employed the standard cross entropy (CE) objective function during training. The NC intervention group described in Section IV did not use a use a CE loss, but instead used a loss function containing the differentiable metrics described below in Section III:

$$L_{NC}(f(x)) = NC_1 + EN_{means} + EN_{classifier} + EA_{means} + EA_{classifier} + NC_3 \tag{2}$$

Note that the metric for $NC4$ is not used, as it requires an *argmin* function which is not differentiable. Although these metrics do not all have the same scale, they all proceed to zero, and we did not find it necessary to use any weighting scheme within the loss function. OoD testing used the CIFAR-10 test set merged with the SVHN test set.

| Intervention | | | NC1 | | NC2 EA Means | | NC2 EA Class | | NC2 EN Means | | NC2 EN Class | | NC3 | | NC4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Spectral Norm | Leaky/GAP | L2 Norm | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| ✗ | ✗ | ✗ | 1.967 | 0.136 | 0.076 | 0.007 | 0.033 | 0.011 | 0.098 | 0.008 | 0.094 | 0.005 | 0.169 | 0.015 | 0.002 | 0.001 |
| | | ✓ | 0.154 | 0.004 | 0.014 | 0.001 | 0.023 | 0.001 | 0.044 | 0.002 | 0.083 | 0.001 | 0.038 | 0.001 | 0.000 | 0.000 |
| | ✓ | ✗ | 2.558 | 0.329 | 0.100 | 0.018 | 0.064 | 0.027 | 0.118 | 0.017 | 0.112 | 0.030 | 0.271 | 0.059 | 0.008 | 0.005 |
| | | ✓ | 0.170 | 0.005 | 0.013 | 0.001 | 0.024 | 0.001 | 0.047 | 0.002 | 0.084 | 0.001 | 0.039 | 0.001 | 0.000 | 0.000 |
| ✓ | ✗ | ✗ | 2.139 | 0.412 | 0.083 | 0.019 | 0.036 | 0.013 | 0.108 | 0.018 | 0.095 | 0.006 | 0.182 | 0.029 | 0.004 | 0.004 |
| | | ✓ | 0.163 | 0.005 | 0.015 | 0.002 | 0.024 | 0.001 | 0.047 | 0.002 | 0.085 | 0.001 | 0.038 | 0.001 | 0.000 | 0.000 |
| | ✓ | ✗ | 2.690 | 0.368 | 0.111 | 0.018 | 0.051 | 0.020 | 0.124 | 0.015 | 0.106 | 0.012 | 0.260 | 0.031 | 0.008 | 0.005 |
| | | ✓ | 0.171 | 0.005 | 0.014 | 0.002 | 0.023 | 0.001 | 0.049 | 0.003 | 0.084 | 0.001 | 0.039 | 0.001 | 0.000 | 0.000 |

Table 3: ResNet18 ablation study: measure of NC metrics across DDU and our method. 15 seeds per experiment, models trained for 60 epochs. Lower indicates more NC. Most metrics are an order of magnitude lower with the L2 intervention.

### 3.2 Measuring Neural Collapse

Papyan et al. (2020) use seven different metrics (Eq. 3 to Eq. 8) to observe properties $NC1$ through $NC4$ described above. All are differentiable and used in the NC loss function for the experiment in Section 4, except for the $NC4$ metric (Eq. 8), which is not differentiable.

The within-class variance, $NC1$, is measured by comparing the within-class variance to the between-class variance,

$$NC1 = Tr\{\Sigma_W \Sigma_B^{\dagger} \backslash C\} \tag{3}$$

where $Tr$ is the trace operator, $[\cdot]^{\dagger}$ indicates the Moore-Penrose pseudoinverse, and $C$ is the number of classes. Although Papyan et al. (2020) could simply have used the trace of $\Sigma_W$ to measure variability collapse, we speculate they incorporated $\Sigma_B^{\dagger}$ (i.e. the between-class precision) because it becomes smaller as the class-means separate from each other around the hypersphere.

$NC2$ is indicated through four measurements. The equinormality of class means and classifier means is given by their coefficient of variation,

$$EN_{means} = \frac{std_c(\parallel u_c - u_G \parallel_2)}{avg_c(\parallel u_c - u_G \parallel_2)} \tag{4}$$

$$EN_{classifier} = \frac{std_c(\parallel w^T \parallel_2)}{avg_c(\parallel w^T \parallel_2)} \tag{5}$$

where $u_c$ and $u_G$ are the class and global means of classes, $std$ and $avg$ are standard deviation and average operators, and $\parallel \cdot \parallel_2$ is the L2 norm operator.

Maximum equiangularity is measured as the mutual coherence of normalized class means or classifier means. Mutual coherence is the Gram matrix of class mean vectors, i.e. the matrix whose elements are the angles between each pair of vectors. Maximum cosine distance between $C$ classes is $-1/(C-1)$ (assuming that the dimension of the embedding space is high enough to embed all vectors at equal angles from each other). We thus add this quantity to each element of the mutual coherence matrix, since all elements will then go to zero as we converge to equiangularity. Finally, we assign zero to the diagonal, as we do not need the distance of vectors with themselves, and we have

$$EA_{means} = \frac{\parallel G + C_{cos} - diag\{G + C_{max}\} \parallel_1}{C * (C-1)} \tag{6}$$

where $G$ is the Gram matrix of normalized class means, $C_{cos}$ is the matrix whose elements are set to $-1/(C-1)$, and $diag$ retrieves the diagonal of a matrix. We take the L1 norm to map this quantity to a single number, and then normalize by the number of pair combinations.

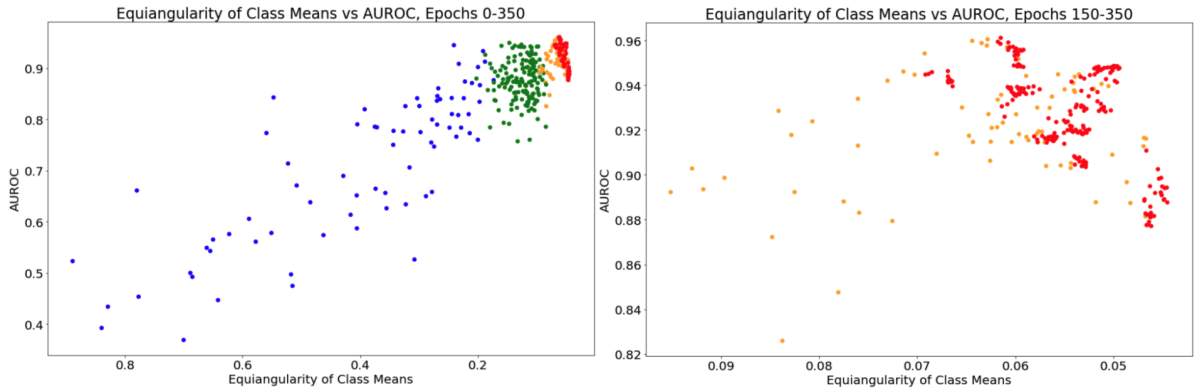


Figure 2: Equiangularity of class means ($NC2$) vs. AUROC scores during training for 15 ResNet18 seeds trained for 350 epochs. Each point is the score of an individual model at a particular epoch, every tenth epoch is captured. Colors are segmented by training epochs: blue 0-40, green 50-140, orange 150-190, and red 190-350. The Pearson R coefficient is -0.894 (NC is decreasing as AUROC increases). Right: only epochs 150-350 are shown (individual models tend to appear as clusters). No correlation exists for this region of training, however, there is evidence of a tightening lower bound with longer training and more neural collapse.

The self-duality of class means and classifiers, $NC3$, is measured as the square of normed differences between classifier and class means,

$$NC_3 = \| \tilde{W}^T - \tilde{M} \|_2^2 \tag{7}$$

where $W$ is the matrix of last-layer classifier weights, $M$ is the matrix of class means, $[\tilde{\cdot}]$ denotes a matrix with unit normed means over columns, and $\| \cdot \|_2^2$ denotes the square of the L2 norm.

Finally, Nearest Class Center classification, $NC4$, is measured as the proportion of training set samples that are misclassified when a simple decision rule based on the distance of the feature space activations to the nearest class is used,

$$NC_4 = argmin_c \| z - u_c \|_2 \tag{8}$$

where $z$ is the feature space vector for a given input.

### 3.3 Regularization of Feature Space

L2 regularization over feature space constrains the feature vector to a point on the surface of a hypersphere in $R^d$,

$$z_{norm} = \frac{z}{max(\| z \|_2, \epsilon)} \tag{9}$$

where $R^d$ is the space of real numbers over $d$ dimensions and $\epsilon$ is added in the event of the zero norm. We note that L2 regularization of the feature space constrains equinormality ($NC2$) from the onset of training. Since feature vector magnitude is no longer a discriminatory factor for the classifier, embeddings must be differentiated entirely by angular distance. This expedites the progression to maximal equiangularity as well as the other aspects of NC (Table 3). Importantly, this can be easily implemented in one line of code and does not require a a sophisticated loss function or parameter tuning.

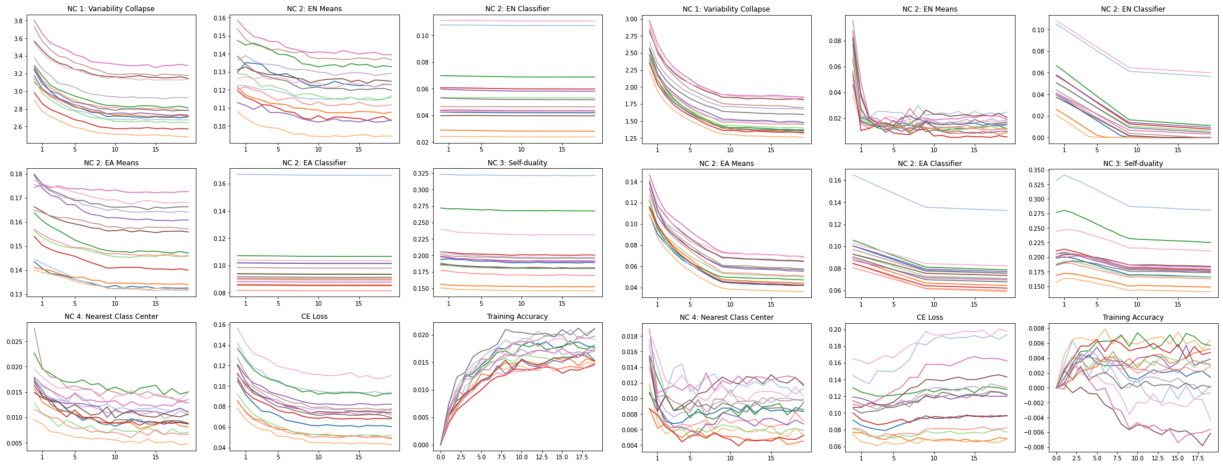


Figure 3: ResNet18 NC measurements, along with CE loss and classification accuracy, epoch of additional training is on all horizontal axes, the particular metric is on the vertical. Left (9 images): control group, Right (9 images): intervention group. In contrast to the control group, the intervention group has substantially more NC, while CE loss and training set classification accuracy are relatively unchanged.

## 4 Experiments

### 4.1 Faster and More Robust OoD Results

Our results in Table 1 demonstrate that L2 regularization over feature space produces results that are competitive with or exceed those obtained with the DDU benchmark, but are obtained in less training time for ResNet18 and ResNet50 models. For ResNet18, our mean AUROC scores exceed those of the baseline in only 60 epochs, with only a .008 reduction in classification accuracy. For ResNet50, we achieve higher mean accuracy than the the baseline in only 100 epochs, while mean AUROC is lower by only 1 point.

Most notably, the lower bounds of OoD detection performance are significantly improved on both models (Table 1). For Resnet18, we improve the AUROC baseline by .047, and on ResNet50 we improve by .024. These effects are even more pronounced when compared with the same number of training epochs in the absence of L2 regularization. Note that while scores could be improved with more training, our scope here is to emphasize that a substantial reduction in compute is possible with this technique. We achieve a substantially better lower bound on OoD performance with only 17-29% of the training time stated in the DDU benchmark. Notably, the variability of AUROC performance is also heavily decreased with L2.

An ablation study shows that mean and minimum accuracy and AUROC are improved across all L2 cases verses those without (Table 2). Furthermore, the lowest minimum and mean scores in the L2 conditions exceed the respective maximum scores for non-L2 conditions.

### 4.2 Improved OoD Detection Under Neural Collapse

All aspects of NC are induced more rapidly under L2 regularization of feature space. Nearly all measures indicate greater NC in 60 epochs than equivalent models trained for 350 epochs. We also note that in models with no L2 regularization, NC and OoD performance are substantially lower after 60 epochs. At first glance, NC appears to be correlated with OoD performance. Figure 2 shows the effect of training non-L2 ResNet18's for 350 epochs on NC and AUROC scores. While there is a general correlation over the course of training (which we would expect, as NC and OoD detection are already known to increase with training), this correlation disappears between epochs 150 and 350. OoD detection can decline with NC, at least near the end of training. There is also evidence of a tightening lower bound as NC and training progress (Figure 2).

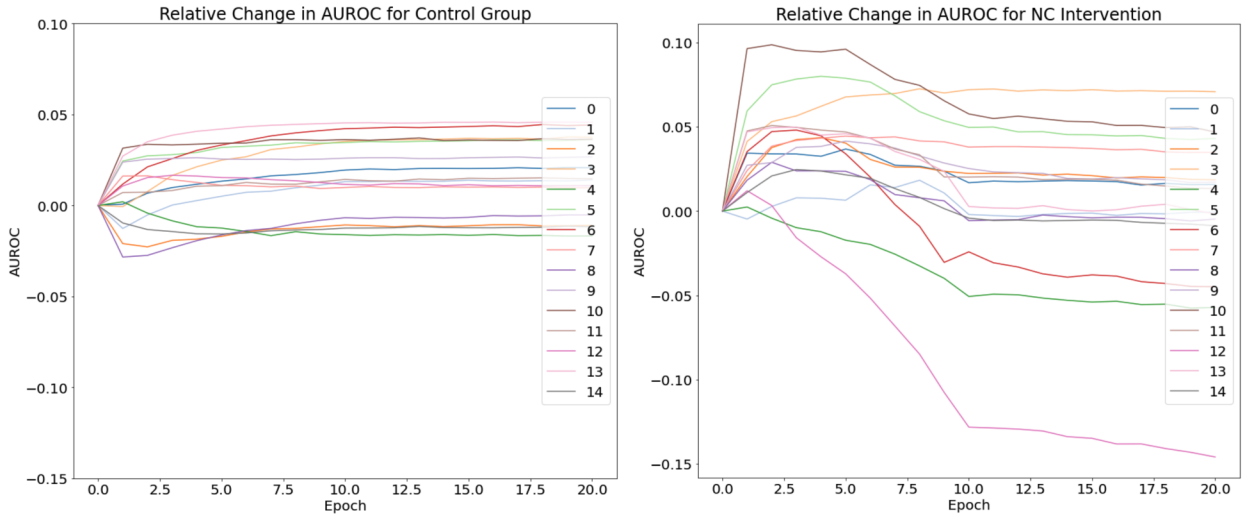


Figure 4: ResNet18 neural collapse control group (left) vs intervention group (right): Relative change in AUROC when training from the 50th epoch over 15 unique model seeds for an additional 20 epochs. The intervention group AUROC score improves an average of 0.38 after two epochs, 4.75x more than the control group mean improvement of .008 over the same period.

In order to investigate the connection between NC and OoD detection further, we use the 15 randomly seeded models from the ResNet18 350 epoch, no-L2 case to create intervention and control groups. We observe that after only 50 epochs of training, accuracy scores and in particular, AUROC scores, approach noisy plateaus, making this a good point for intervention. We continue training all models from the 50th epoch using the differentiable NC metrics listed above (the intervention group) and separately again with standard cross entropy (the control group). As we had earlier observed that the first learning rate step down to $1e^{-2}$ (at epoch 150) results in increases to both AUROC and accuracy, we start the learning rate for the intervention and control groups an order of magnitude lower ($1e^{-2}$) than its initial setting ($1e^{-1}$), and then step down to $1e^{-3}$ after 10 epochs in order to control for a possible effect from learning rate.

As shown in Figure 3, models in the intervention group have substantially greater amounts of NC, as would be expected. CE loss and classification accuracy remain relatively stable in the intervention models, owing to the fact that CE is excluded from the loss function. The intervention group is thus largely disentangled from confounding influence from the learning rate or the CE objective.

In Figure 4, we see that the effect of NC is substantially different between intervention and control groups as measured by OoD detection performance. Models perform better under NC and these improved scores arrive with less training: after only 2 epochs, intervention models improve AUROC an average of 0.038 points from intervention onset, 4.75x the average improvement of .008 from control models at during this time. Even if allowed to train for the full 20 epochs, control models improve to a maximum average of .017.

Notably, more NC is not always better. Nearly all models begin to perform worse than their peak after 5 epochs of direct NC training (Figure 4). Moreover, we observe that no amount of NC–by individual measure or in total–is directly correlated with AUROC. While it is clear that NC has a regularizing effect on OoD detection performance, the precise mechanism at play is unclear and is a subject of future study.

## 5 Conclusion and Future Work

We propose a simple, one-line-of-code modification of the proposed Deep Deterministic Uncertainty benchmark Mukhoti et al. (2021) that provides near or superior OoD detection and classification accuracy results in a fraction of the training time. We also establish that L2 regularization induces NC faster than regular training, and that NC encourages OoD detection performance under this method. Although we do not

suggest that NC is the sole or even a comprehensive explanation for OoD performance, we do expect that its simple structure can provide insight into the complex and poorly understood behaviour of uncertainty in deep neural networks. We believe that this connection is a compelling area of future research in topics of uncertainty and robustness.

## References

Jiefeng Chen, Yixuan Li, Xi Wu, Yingyu Liang, and Somesh Jha. Robust out-of-distribution detection in neural networks. *CoRR*, abs/2003.09711, 2020. URL `https://arxiv.org/abs/2003.09711`.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Jarrod Haas and Bernhard Rabus. Uncertainty estimation for deep learning-based segmentation of roads in synthetic aperture radar imagery. *Remote Sensing*, 13(8), 2021. ISSN 2072-4292. doi: 10.3390/rs13081472. URL `https://www.mdpi.com/2072-4292/13/8/1472`.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *CoRR*, abs/1812.05720, 2018. URL `http://arxiv.org/abs/1812.05720`.

Maximilian Henne, Adrian Schwaiger, Karsten Roscher, and Gereon Weiss. Benchmarking uncertainty estimation methods for deep learning with safety-related metrics. In *SafeAI@ AAAI*, pages 83–90, 2020.

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.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

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. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):1–13, 2022.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *CoRR*, abs/2102.11582, 2021. URL `https://arxiv.org/abs/2102.11582`.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Stephan Rabanser, Stephan Günnemann, and Zachary C. Lipton. Failing loudly: An empirical study of methods for detecting dataset shift, 2018. URL `https://arxiv.org/abs/1810.11953`.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.

Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. *arXiv preprint arXiv:2205.09310*, 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*, 34:29820–29834, 2021.