Exploring Simple, High Quality Out-of-Distribution Detection with L2 Normalization

Anonymous authors
Paper under double-blind review

Abstract

We demonstrate that L2 normalization over feature space–an extremely simple method requiring no additional training strategies, hyperparameters, specialized loss functions or image augmentation–can produce competitive results for Out-of-Distribution (OoD) detection. It requires only a fraction of the training time (60 epochs with ResNet18, 100 epochs with ResNet50) of more sophisticated methods. We show theoretically and empirically that our simple method decouples feature norms from the Neural Collapse (NC) constraints imposed by CE loss minimization. This decoupling preserves more feature-level information than a standard CE loss training regime, and allows greater separability between ID norms and near-OoD or far-OoD norms. Our goal is to provide insight toward fundamental, model-based approaches to OoD detection, with less reliance on external factors such as hyperparameter tuning or specialized training regimes. We suggest that L2 normalization provides a collection of benefits large enough to warrant consideration as a standard architecture choice.

1 Introduction

Neural networks give unreliable confidence scores when processing images outside their training distributions, hindering applications in real-world settings where reliable failure modes are critical (Rabanser et al., 2018; Chen et al., 2020; Henne et al., 2020). Much work has been done to address this shortcoming. Although Monte-Carlo Dropout (MCD) and model ensembles were initially popular (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017), they have quickly become obsolete. A large, diverse landscape of approaches has replaced these simpler and well-studied methods, and has improved OoD detection substantially (see Section 2.2). While this growth is remarkable, the complexity of many recent methods can be intimidating, and it is not always clear whether performance improvements are due to external factors such as introducing new or enlarging existing models, hyperparameter tuning, training schedules or augmentation schemes. In contrast, we propose L2 normalization over feature space during training as a simple, parameter-free architectural change that affords a growing body of theoretical insight, faster and straightforward training, and competitive performance.

L2 normalization over feature space has been utilized in the facial recognition literature for several years but has only recently been explored more generally with respect to OoD detection (Regmi et al., 2023; Park et al., 2023; Yu et al., 2020) and training efficiency (Haas et al., 2023; Yaras et al., 2023). We build upon this previous work, demonstrating that L2 normalization results in test-time feature vector magnitudes that can perform competitively with state-of-the-art methods for OoD detection. We observe that test-time magnitudes are not competitive for OoD detection without the use of L2 normalization during training, and show that this is due to Neural Collapse (NC) that results from training with Cross-Entropy (CE) loss. L2 normalization greatly increases the dynamic range of feature vectors due to a decoupling of their magnitude and direction when determining confidence scores and meeting optimal loss conditions during training. Feature norms grow larger with changes to weights that affect them during gradient descent. This results in a greater degree of feature-level information encoded in feature norms: intuitively, larger norms indicate that the network is more familiar with the input.

A summary of our contributions is as follows:

- A simple, competitive method for OoD detection, that requires no additional parameters and works with standard architectures (e.g. ResNet, VGG) and training regimes (standard augmentations, SGD with a simple learning rate schedule)
- A method that is efficient, requiring far less training (< 100 epochs) compared to others' recent methods, some of which report 350 - 800 epochs of training for a ResNet18 / CIFAR10 configuration
- A theoretical rationale for the connection between L2 normalization of features, Neural Collapse and the usefulness of feature magnitudes for OoD detection

2 Background and Methodology

2.1 Problem Setup

A standard classification model maps images to classes $f : x \to \hat{y}$, where $x = (x_1, ..., x_n) \in X^N$ is the set of N images, and $y = \{1, ..., k\} \in Y^N$ is the set of labels from k classes for these images. The model is composed of a feature extractor (or encoder) $H : x \to \mathbb{R}^d$ which maps images to feature space vectors $z = (z_1, ..., z_n) \in Z^N$, with d being the size of the vector. Along with a decision layer $W^{d \times k}$ and a bias term b, this becomes an optimization problem of the form:

$$\min_{W,b,z} \frac{1}{N} \sum_{N} L_{CE}(Wz + b, y_k) \tag{1}$$

where y_k is the label of the correct class, and L_{CE} is the Cross Entropy loss function. Note that the logits $Wz + b$ are transformed by the softmax function for the entropy computation. For analysis, we consider the encoder H as a black box that produces features z which collapse to a Simplex Equiangular Tight Frame (ETF), as in the Unconstrained Features Model (UFM) or Layer-Peeled Model (LPM) (Mixon et al., 2022; Ji et al., 2021). We use VGG16, ResNet18 and ResNet50 as encoders and specify the number of training epochs used where needed (See Section A.1 for more details). If type is omitted when referring to a model, e.g. NoL2 350, we mean a ResNet18 trained without L2 normalization for 350 epochs. Models trained with L2 normalization over features are tested without it.

To evaluate models, we merge ID and OoD images into a single test set. OoD performance is then a binary classification task, where we measure how well OoD images can be separated from ID images using a score derived from our model. Our score in this case is the L2 norm of each image's *unnormalized* feature vector z, which is input to AUROC and FPR95 scoring functions.

2.2 Motivation and Related Work

A large number of OoD methods have been published in recent years, and no standard taxonomy for these exists. We situate our work within this rapidly evolving and expanding landscape of research, which includes several categories:

- **Bayesian** - Concerned with approximations of Bayesian statistics to deep learning, such as Monte Carlo Dropout, Probabilistic Back-Propagation, mean-field approximations, variational inference methods
- **Outlier exposure** - Requiring real or synthetic OoD samples during training or tuning of metrics
- **Density/Distance methods** - Analyzing feature space using density estimation (e.g. normalizing flows, Gaussian Mixture Models (Mahalanobis distance), or using non-parametric methods such as euclidean distance or KNNs

- **Generative models** - Learning an implicit (e.g. GANs) or explicit (e.g. normalizing flows) distribution over the data set, and exploiting this to evaluate the probability of samples
- **Augmentation** - Changing input data before model ingestion during training or inference
- **Measurement** - Using parts of the network other than softmax outputs to obtain a confidence score, or making alterations to softmax outputs to obtain a confidence score
- **Architectural** - Changing standard architectures, such as ResNet or Transformers, to affect how confidence scores are obtained

Many recent methods fall into multiple categories and are an admixture of techniques. Although many of these methods report very high performance, architectures can vary substantially or require very specific or sophisticated hyperparameter tuning and training strategies. This makes it difficult to distinguish fundamentals from externalities or "tricks." This difficulty in assessing real progress is not unique to the OoD research community, and has already emerged as a concern within the metric learning community (Musgrave et al., 2020).

Our motivation is to provide a method that 1) is simple and yet competitive, 2) requires as little divergence as possible from standard architectures and training strategies and 3) has theoretical and empirical evidence that situates it as a possible fundamental design choice. L2 normalization requires only a simple architectural change, and is supported by evidence that it can speed up feature learning, improve accuracy and improve worst case performance for OoD tasks. We believe that this combination of efficiency, performance, simplicity and theoretical insight supports our goals. We do not aim to provide a comprehensive comparison of all recent state-of-the-art methods or to demonstrate how our method may or may not improve with respect to the aforementioned externalities.

Several papers have explored concepts related to L2 normalization and OoD detection. The face verification community has used L2 normalization over feature space, noting that it can improve accuracy and face-matching quality during training. Ranjan et al. (2017) use a scaled L2 normalization layer over features, but provide no insight as to why this improves performance. Wang et al. (2017b) produce a similar idea, and note that L2 normalization over features can produce arbitrarily large gradients without weight decay (an idea which we borrow), but view this more as a side-effect that needs to be mitigated than a feature. Yu et al. (2020) use L2 norms of features as a score to detect non-face OoD samples, but implements a modified softmax loss with an "uncertainty branch" and offers no insight into why the method works.

Yu et al. (2023) show that using the norm of intermediate network layer outputs (after training) can provide good far-OoD separability (the method does not work well with near-OoD data), and use pixel-wise shuffled training data to assess which layer in a particular network may be most suitable. Wei et al. (2022) achieve good results by L2-normalizing *logits* during training, in conjunction with a temperature parameter. Finally, Regmi et al. (2023) concurrently proposed the same method as ours. However, we note that none of these give explanations of why their methods provide improved OoD separability.

Vaze et al. (2022) briefly offer intuition of why feature norms grow larger for ID examples (larger norms lower CE loss values; smaller norms hedge against incorrect predictions in hard or uncertain examples). However, this is is an afterthought in their appendix. Their main contribution is exploiting extreme over-training (600 epochs), augmentation strategies, and learning rate schedules for open-set recognition. We also note that feature norms *do* grow larger for ID examples. However, this effect is subtle and not competitive as a scoring rule on its own, and we explain why theoretically and demonstrate this experimentally.

Park et al. (2023) provide some theoretical analysis as to why feature norms are useful as OoD detectors. Their two main arguments are that feature norms of layers are equivalent to confidence scores, and that feature norms tend to be larger for ID examples, regardless of class. However, their analysis again depends on extreme over training with a cosine annealing schedule (800 epochs for ResNet18 under CE loss), requires the addition of a multi-layer perceptron (MLP) head (with a much larger output size) to the encoder, requires a temperature parameter, and does not connect L2 normalization with neural collapse and feature norm behaviour. We note that adding l2 normalization to the output of an MLP head placed on top of the

Method	Network	SVHN	CIFAR100
Mahalanobis	ResNet34	99.1	90.5
LogitNorm+	ResNet18	97.1	91.0
FeatureNorm	ResNet18	97.8	74.5
KNN+	ResNet18	99.5	N/A
SSD+	ResNet50	99.9	93.4
CSI (Unsupervised)	ResNet18	99.8	89.2
Gram	ResNet34	99.5	79.0
Contrastive (SimCLR + Mahalanobis)	WideResNet50_4	99.5	92.9
ERD++	3x ResNet20	1.00	95.0
Pretrained Transformer + Mahalanobis	R50+ViT-B_16	99.9	98.5
L2 Mag 60	ResNet18	97.4	88.7
L2 Mag 100	ResNet50	98.6	90.4

Table 1: Summary of our results for models trained on CIFAR10 and tested on far OoD (SVHN) and near OoD (CIFAR100) data sets. We include results with similar architectures where possible. Although ERD and transformers require multiple models and many more parameters respectively, we include them to show where the current SoTA is. Our method is competitive with more complex methods, while requiring very little training time and no additional hyperparameter tuning, augmentation, test-time computation or changes to the loss function. We anticipate that sophisticated training recipes that include processes such as data augmentation (e.g. mix-up, cut-mix) or contrastive loss would improve our results.

encoder is a fundamentally different analysis from encoder output that immediately precedes a decision layer (for more details, see Ji et al. (2021), Papyan et al. (2020) and Mixon et al. (2022)).

2.3 Cross Entropy Loss and Neural Collapse

Papyan et al. demonstrated that Neural Collapse occurs in feature space for models given an encoder, decision layer and CE loss function. We are concerned with the first two properties of NC:

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

NC2: Convergence to a Simplex Equiangular Tight Frame (Simplex ETF): the angles between each pair of class means are maximized and equal (equiangularity) and the distances of each class mean from the global mean of classes are equal (equinormality), i.e. class means are placed at maximally equiangular locations on a hypersphere.

Shortly afterward Graf et al. (2023) anaylzed CE loss and it's lower bound:

$$L_{CE}(Wz + b, y_k) \geq log\left(1 + (k-1)\exp(-\rho_z \frac{\sqrt{k}}{k-1}||W||_F)\right) \tag{2}$$

where $||\cdot||_F$ is the Frobenius norm, $\rho_z \geq 0$, and $||z||_2$ is less than ρ_z. Equality is possible if and only if features collapse to k equal length, maximally equiangular class vectors (see Figure 1) and the k decision layer columns in W form a dual space with class vectors up to a scalar.

Since encoders optimized under CE loss will produce feature vectors of equal length and direction for each class, this means that features will tend to contain *class-level* information at the expense of *feature-level* information. Intuitively, this is unsurprising given that the classification task optimized under CE loss is simply to categorize images and not to articulate intra-class differences. While this scheme has proven expedient for image classification, it *a priori* indicates tasks such as uncertainty measurement or OoD

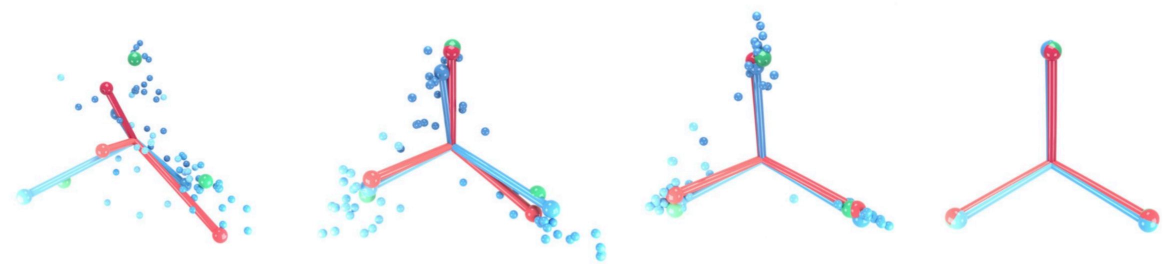

Figure 1: Progression of Neural Collapse during training (left to right). Under Cross Entropy loss, features converge to equinormal and equiangular vectors. Small blue spheres represent extracted features (classes are different shades of blue), blue ball-and-sticks are class-means, red ball-and-sticks are linear classifiers. The simplex ETF pictured is on the 2D plane in 3D space, such that each arm is equidistant at 120 degrees. Image from (Papyan et al., 2020).

detection are likely to be problematic. We show that this is indeed the case in Section 3. In the next section we explore how L2 normalization decouples feature magnitude from the CE optimization process and provides a means to preserve more intra-class information relevant to OoD detection.

2.4 Decoupling Feature Norms from CE Loss with L2 Normalization

L2 normalization has emerged as a useful tool for its role in feature quality, OoD detection (see Section 2.2, and faster training (Haas et al., 2023; Yaras et al., 2023). Yaras et al. (2023) analyzed the optimization of CE loss under L2 normalization of features and decision weights and noted that this configuration converges to NC. They also noted that L2 normalization increases NC.

L2 normalization over features is defined as:

$$z_{norm} = \frac{z}{max(\|z\|_2, \epsilon)} \tag{3}$$

where ϵ is an error term for numerical stability and z is the output of the encoder. Wang et al. (2017b) pointed out that L2 normalization during training had the potentially problematic side-effect of allowing feature magnitudes to grow arbitrarily large in the absence of weight decay. That is, when features are normalized, the gradient of the loss w.r.t. features is orthogonal to the features:

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial z_{norm}} \frac{\partial z_{norm}}{\partial z}, \qquad \left\langle \frac{\partial L}{\partial z}, z \right\rangle = 0 \tag{4}$$

where L is any loss function and $\langle\rangle$ denotes an inner product.

The proof of Equation 4 can be found in (Wang et al., 2017a). As a result, all changes to features caused by gradient adjustments to weights during the backward pass push features along their tangent line to the hypersphere. This makes features larger (just as the hypotenuse grows larger with side length, as per Pythagorean theorem, see Figure 2). During the subsequent forward pass features are normalized, preventing the effect of this increase from affecting the loss, although any change in direction is preserved. This means that feature magnitudes are partially decoupled from the loss: magnitudes become correlated with total change due to backpropagation, but these changes in magnitude do not impact model confidence.

We view this tendency for norms to grow as a feature, not a bug. Intuitively, features that grow large due to change from back propagation reflect a measure of how much the network has had to adjust weights that affect the feature representation of an image. This affords the possibility of richer feature-level information which CE loss discards or ignores due to the neural collapse traits present in its optimization behaviour. Since training efficiency is included for free and no additional hyper-parameter tuning is necessary, we believe

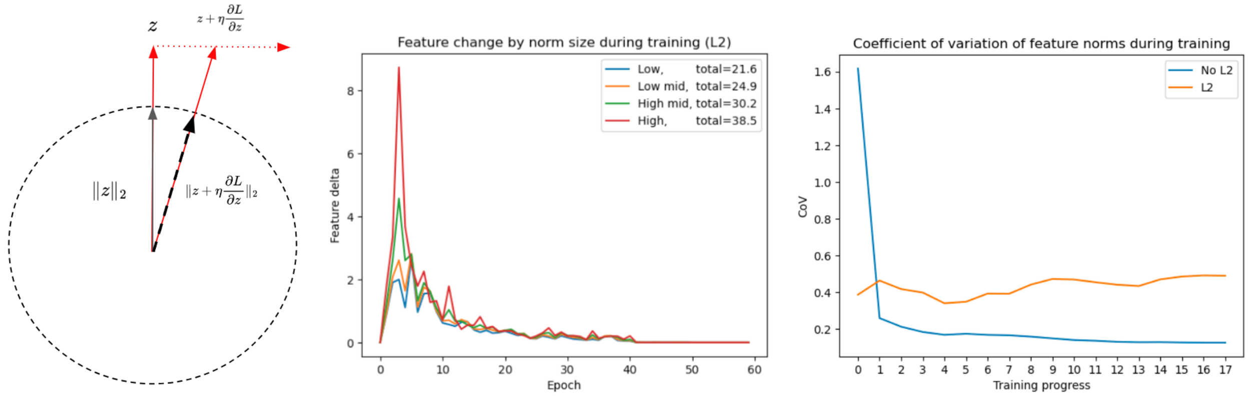

Figure 2: (Left) As a result of the orthogonality of the loss w.r.t. features (when L2 normalization is used), any weight updates that result in changes to features push features along their tangent line to the hypersphere. This means each backward pass makes features slightly larger, and this trend continues indefinitely in the absence of weight decay (Image adapted from (Wang et al., 2017a)). (Center) We separate converged features of the CIFAR10 test set into four groups, based on feature norms. As predicted, features that had the highest magnitudes at the end of training change the most during training. (Right) Variability of norms decreases during training to minimize CE loss without L2 normalization (the equinormality condition of NC), however, it increases during training with L2 normalization.

L2 normalization (and possibly other forms of normalization) is worthy of consideration as a fundamental design decision when training neural networks.

3 Experiments

Our results in Table 1 demonstrate that L2 normalization over feature space during training produces results competitive with state-of-the-art methods. Notably, L2 normalization results not only in large performance gains over non-normalized baselines, but these gains happen much faster (Table 2, Table 3). Similar training efficiencies were noted previously in (Haas et al., 2023) as well as (Yaras et al., 2023). Training details for all experiments can be found in A.1.

3.1 Analysis of Norm Decoupling

Given that the most confident predictions have the largest feature norms after training, (with L2 normalization off, Figure 3), Equation 4 predicts that these features should also experience the most change during training. Specifically, due to the orthogonal gradients of the pre-normalized features with respect to the loss, features that are pushed off the hypersphere the most due to weight updates throughout training should also become the greatest in size.

To test this, we use a converged L2 60 model. CIFAR10 training images were separated by their feature norms into four categories: low in [0,10), low-mid in [10,20), high-mid in [20,30) and high in [40, ...]. These images were then run through saved model checkpoints at each epoch, and respective total changes of each group were calculated from one epoch to the next and plotted. Total change Δ_m per magnitude grouping m was then calculated as the mean of the squared difference of features at consecutive epochs during training:

$$\Delta_m = \sum_i \overline{(z_{m,i} - z_{m,i-1})^2} \tag{5}$$

where i is the epoch of the model checkpoint and $z_{m,i}$ is the collection of feature vectors for size group m at epoch i. As expected, features with more change during training have larger magnitudes (Figure 2). We observe that this behaviour is not present when training without L2 normalization (Figure 5).

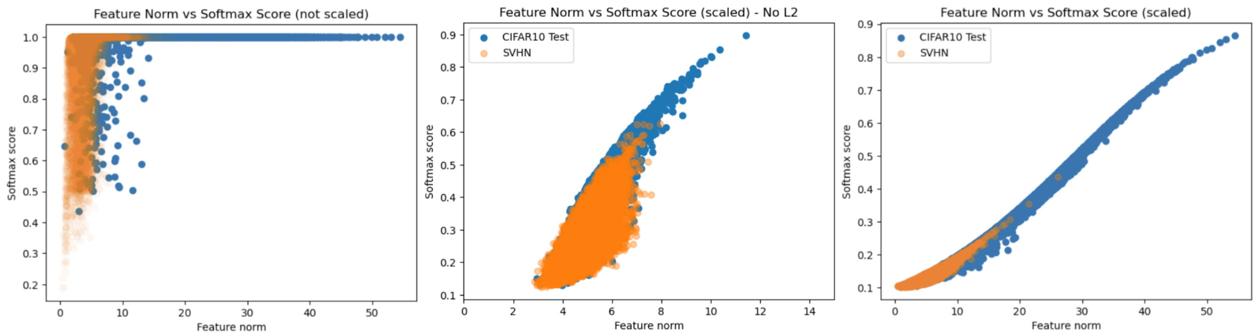

Figure 3: Plots of feature norms vs softmax scores for all CIFAR10 (blue) and SVHN (orange) test images. (Left) Allowing feature magnitudes to grow saturates the softmax function, and decreases the headroom available for separability. This occurs in both L2 and NoL2 models, although saturation is greater in L2 models. (Center) Scaled down features from the NoL2 ResNet18 350 model. (Right) Scaled down features from the L2 ResNet18 60 model. The nearly linear correlation of feature norm to softmax, in addition to the more isolated cluster of OoD images, is evidence that more feature-level information is retained on a per-image basis than in NoL2 models.

Notably, unnormalized feature norms become tightly correlated with model confidence (provided that the softmax saturation caused by these larger norms is first corrected (by downscaling prior to ingestion by the decision layer). Feature norms provide a much more uniform coverage of softmax space than their counterparts without L2 (Figure 3). This suggests that feature-level information in images is not being collapsed into class-level information, or that this is happening to a much smaller degree. Norms also have a nearly perfect monotonic relationship with accuracy (Figure 6), further supporting the claim that feature norms become a reliable estimate of the model's familiarity with the input under L2 normalization.

While it is clear that larger norms would produce higher confidence scores, it is unclear why the amount of change to features over training correlates so strongly with model confidence *and accuracy*. Equation 4 predicts that features that change the most during training would have the largest norms at convergence. In the absence of L2 normalization during training, larger features mean a larger dot product with the decision layer, and thus a larger softmax score. Although neural collapse dictates that norms trend to equinormality, there are still subtle variations. However, *with* L2 normalization during training, magnitude is factored out of the features for the decision layer dot product–only direction and size of the decision layer weights is relevant. Since decision layer weights become a dual space of features (Papyan et al., 2020), decision boundaries also asymptotically approach equinormality and maximal equiangularity.

We speculate some combination of 1) features that have moved around more are more likely to be closer to decision points because more effort has been invested in placing them there and 2) "common" features exist (i.e. features that provide strong class signals across many images) that have their respective weights updated more often because of their presence across so many batches. Intuitively, it is likely that images with low confidence also have rare and/or noisy features, and updates to the respective weights simply participate in backpropagation less often and in a less prominent manner. A causal analysis of the exact mechanism is challenging and left for future work.

3.2 Effect of the Equinormality Constraint in CE Loss

As predicted by the optimization analyses in (Graf et al., 2023; Yaras et al., 2023), variability of feature norms decreases (i.e. to meet equinormality conditions) during training in order to minimize loss for NoL2 models (Figure 2). Variability is far greater in the L2 case, and increases as training progresses. Feature activations become much more sensitive to ID data than to OoD data of various kinds (Figure 4).

Even though there is a clear mechanism that makes feature norms larger and more variable during training under L2 normalization, the question still remains as to why OoD detection works much more poorly without it. One might speculate that, even though CE loss has an equinormality condition, OoD features would still

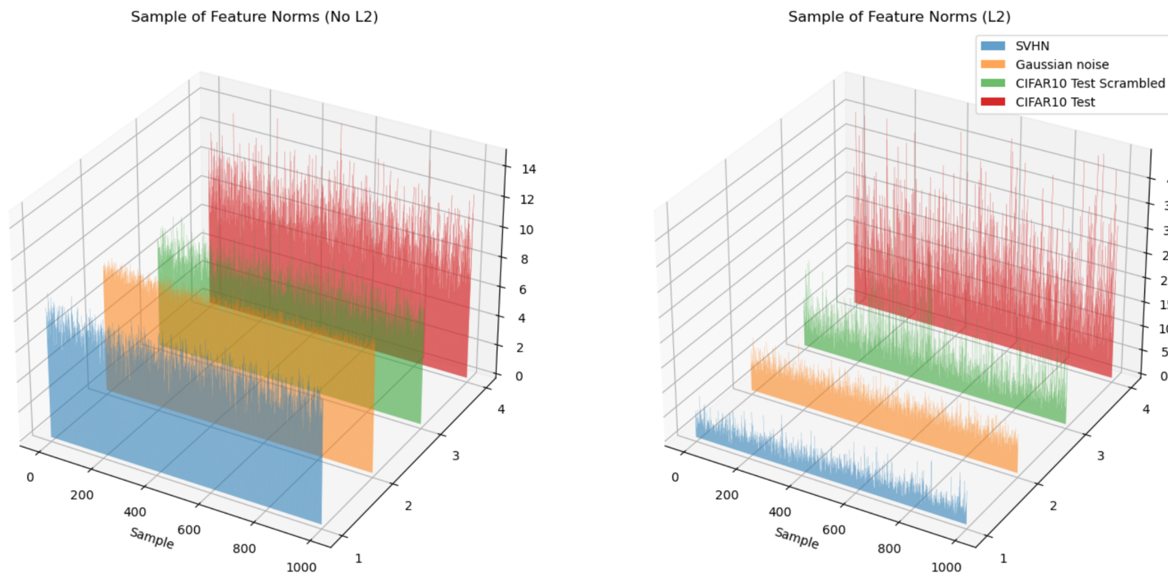

Figure 4: Comparison of norms under NoL2 ResNet18 350 (left) and L2 ResNet 60(right) models, four datasets (CIFAR10 Test, SVHN, Gaussian noise and pixel-wise scrambled CIFAR10 Test). We hypothesize that models without L2 normalization generate norms that are invariant to inputs in order to meet the equinormality condition of CE loss. Our results support this claim: all datasets have roughly equivalently sized norms on average with very little variation amongst them. When L2 normalization is used during training, ID norms (measured prior to normalization) grow large and have high variability, while OoD norms are much smaller. The sensitivity of convolutions to features is not being conditioned/suppressed to produce equinormality in the latter case, which results in richer feature-level information and better OoD detection.

activate convolutions to a smaller degree, and thus generally result in smaller feature norms. The results in Table 2 show that this is somewhat the case: separability is well above random chance (particularly with ResNet50) but not at all competitive. We believe that this still comes down the equinormality condition of CE loss. Specifically, we hypothesize that equinormality is achieved largely through norm invariance to inputs: the network develops convolutions in such a manner as to generate similar sized feature norms under any image.

To test this, we compare feature norms from CIFAR10, SVHN, pixel-scrambled CIFAR10, and Gaussian noise. The results are shown in Figure 4. In NoL2 models, norms are almost the same average size across all datasets and variability within datasets is limited. This changes markedly in L2 models, where ID has not only more variation in amongst norms, but also much larger norms on average for ID data. This provides evidence that models attempt to produce invariant norms without L2 normalization over features, and that this effect is removed under L2 normalization as per Section 2.4.

3.3 Measuring Decoupled Feature Norms

The question of how best to measure this increased, feature-level information remains. Several papers, beginning with Hendrycks et al. (2019), report strategies based on taking the maximum logit (MaxLogit) (Vaze et al., 2022; Jung et al., 2021; Zhang and Xiang, 2023). Measuring the norm of the logits or the maximum softmax score are also options.

Softmax measurements are problematic for two reasons. First, the model is incentivized to increase weight size in the decision layer to reduce loss, as per Equation 2. This increases saturation of softmax outputs (see Figure 3), since logits are a dot product of feature vectors with the columns of decision layer weights. Secondly, in order to maximize the benefits of feature magnitudes, normalization must be turned off at test time. Substantially larger features easily saturate the softmax, removing much of the dynamic range needed

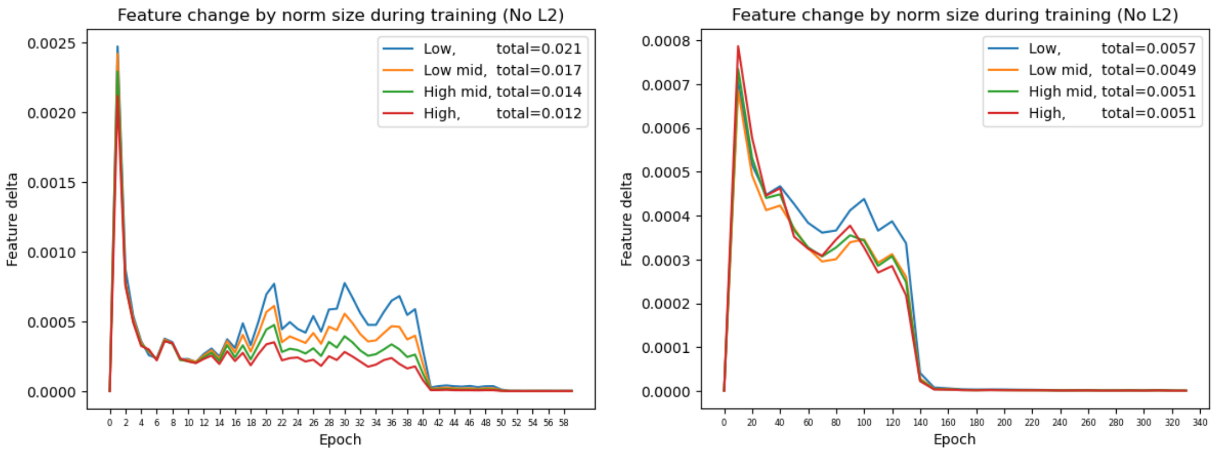

Figure 5: Feature change for four categories of norms in NoL2 models: low in [0,5), low-mid in [5,7), high-mid in [7,9) and high in [9, ...]. Note that these ranges differ from the L2 models, as those norms are much larger (see Section 3.1). (Left) results from a NoL2 ResNet18 60 model, (Right) results from a NoL2 ResNet18 350 model. Unlike the L2 models, a larger change in features during training does not result in larger feature norms. In the 60 epoch model this trend is reversed, while in the 350 epoch model change per category is roughly equal. Much more neural collapse occurs with the longer training regime, so it is likely that this equality is a reflection of the equinormality constraint of CE loss).

to separate OoD examples. A few options exist for mitigating softmax saturation, including normalization of decision layer weights, weight decay and feature scaling.

Scaling these features down (which requires a post-hoc estimate of the optimal scaling value) produces a nearly linear correlation between softmax scores and feature magnitudes. This greatly improves OoD separation with softmax scores, but measuring magnitudes directly still gives a very small advantage (Table 4). While interesting, we don't see any practical use for this approach, and it does not improve accuracy. We also experimented with using L2 normalization over decision weights and larger weight decay values for the optimizer with the hope of mitigating saturation, but these required more training time and resulted in worse performance. In our estimation, the best approach is to measure the feature norms directly. This is the most direct source of information about the inputs. Neural collapse stipulates that the decision layer will form a dual space (up to a scalar) of the simplex ETF in feature space (Papyan et al., 2020): taking measurements from the decision layer and lower should add no substantial information, as is supported by our experimental results. Furthermore, we see no benefit to the additional complexity added by adjusting features or logits with scaling or softmax temperature terms.

4 Conclusion

Our results in Table 1 demonstrate that L2 normalization over feature space during training produces OoD detection results that are competitive with state-of-the-art methods. We provide a theoretical rationale for this behaviour, and observe that this is due to the inclusion of more feature-level information into feature norms than is afforded in standard CE loss training regimes. Notably, this method requires less training, nominal additional computation, no complex architectures or loss functions, no extra hyperparameter tuning and no augmentation schemes–though the addition of these could further improve results. We suggest that the nexus of improvements offered by L2 normalization makes it (or perhaps other normalizations) a candidate for standard inclusion in neural network architectures and a promising avenue for further study.

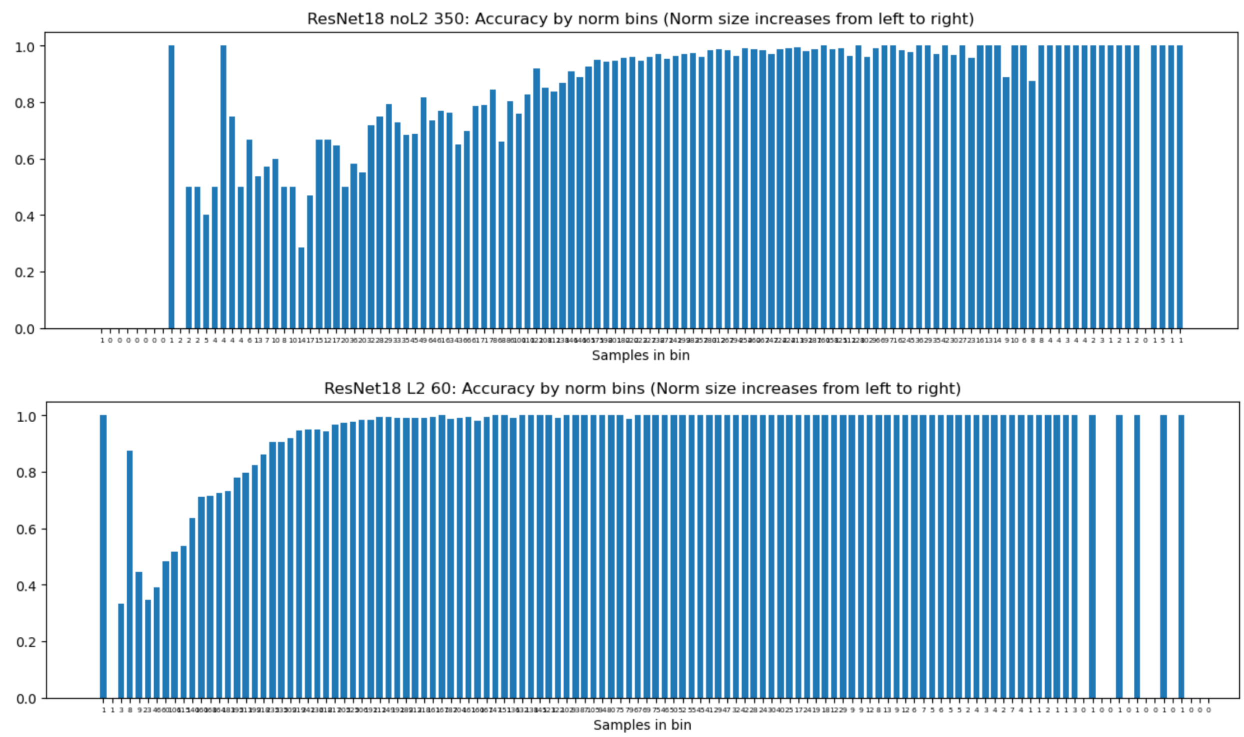

Figure 6: Feature norm vs. prediction accuracy for ResNet18 models with and without L2 normalization on CIFAR10 test data. Test images are grouped into 125 equal sized bins, the accuracy of each bin is represented by the height of the bar and the number of images in the bin is demarcated on the x-axis. Norm size increases in uniform increments from left to right. A missing bar indicates no feature norm exists for that bin. (Top) accuracy tends to improve with norm size for the NoL2 350 model, but the trend is noisy. (Bottom) aside from nine anomalous images with very small norms, there is a nearly perfect monotonic relationship between norm size and accuracy. It is not clear why these images nine images behave so differently from the others.

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`.

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.

Florian Graf, Christoph D. Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised contrastive learning, 2023.

Jarrod Haas, William Yolland, and Bernhard T Rabus. Linking neural collapse and l2 normalization with improved out-of-distribution detection in deep neural networks. *Transactions of Machine Learning Research*, 2023.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings, 2019.

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.

Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation, 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.

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`.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check, 2020.

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.

Jaewoo Park, Jacky Chen Long Chai, Jaeho Yoon, and Andrew Beng Jin Teoh. Understanding the feature norm for out-of-distribution detection, 2023.

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`.

Rajeev Ranjan, Carlos D. Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification, 2017.

Sudarshan Regmi, Bibek Panthi, Sakar Dotel, Prashnna K. Gyawali, Danail Stoyanov, and Binod Bhattarai. T2fnorm: Extremely simple scaled train-time feature normalization for ood detection, 2023.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: a good closed-set classifier is all you need?, 2022.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. Normface: L_2 hypersphere embedding for face verification. *CoRR*, abs/1704.06369, 2017a. URL `http://arxiv.org/abs/1704.06369`.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. NormFace. In *Proceedings of the 25th ACM international conference on Multimedia*. ACM, oct 2017b. doi: 10.1145/3123266.3123359. URL https://doi.org/10.1145%2F3123266.3123359.

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.

Can Yaras, Peng Wang, Zhihui Zhu, Laura Balzano, and Qing Qu. Neural collapse with normalized features: A geometric analysis over the riemannian manifold, 2023.

Chang Yu, Xiangyu Zhu, Zhen Lei, and Stan Li. Out-of-distribution detection for reliable face recognition. *IEEE Signal Processing Letters*, PP:1–1, 04 2020. doi: 10.1109/LSP.2020.2988140.

Yeonguk Yu, Sungho Shin, Seongju Lee, Changhyun Jun, and Kyoobin Lee. Block selection method for using feature norm in out-of-distribution detection, 2023.

Zihan Zhang and Xiang Xiang. Decoupling maxlogit for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3388–3397, June 2023.

A Appendix

A.1 Training Details

All experiments use fifteen standard ResNet18 models initialized randomly and trained with Cross Entropy (CE) loss on CIFAR10.

All models use an SGD optimizer initialized to a learning rate of 1e-1 with gamma=0.1, and with stepdowns at 40 and 50 epochs for 60 epoch models, 75 and 90 epochs for 100 epoch models, and at 150 and 250 epochs for 350 epoch models. All of our models use spectral normalization, global average pooling, and Leaky ReLUs, as these have shown to produce strong baseline performance (Mukhoti et al., 2021; Haas et al., 2023). We apply these strategies to LogitNorm, and demonstrate a small improvement, so we report the stronger baseline which we call LogitNorm+ (Table 1). L2 normalized models were trained with a batch size of 1024, but LogitNorm models were trained with a batch size of 128 as per the original paper's recommendations (Wei et al., 2022). All other baselines are reported as per their respective papers.

A.2 Additional CIFAR10 Results

	SVHN		**CIFAR100**		**TinyImageNet**		**Accuracy**	
Method	NoL2	L2	NoL2	L2	NoL2	L2	NoL2	L2
VGG16	82.4 / 72.2	95.8 / 28.5	84.4 / 63.1	88.4 / 59.1	86.4 / 57.0	89.9 / 50.9	92.4	92.0
ResNet18	77.8 / 76.6	97.4 / 14.8	81.0 / 66.5	88.7 / 55.4	80.5 / 67.3	91.1 / 44.6	93.7	92.6
ResNet50	85.8 / 51.7	98.6 / 7.80	77.7 / 66.1	90.4 / 49.3	77.8 / 64.3	91.9 / 43.8	93.8	94.1

Table 2: Comparison of models under L2 intervention for AUROC/FPR95 scores, using feature norms as the scoring rule. All NoL2 models are trained for 350 epochs, while L2 VGG16 and L2 ResNet18 are trained for 60 epochs, and ResNet50 is trained for 100 epochs. L2 models produce substantially better OoD results. Notably, accuracy scores are nearly as good, even when NoL2 models are trained for much longer, indicated in increase in training efficiency under L2 normalization.

	SVHN		**CIFAR100**		**TinyImageNet**		**Accuracy**	
Model	NoL2	L2	NoL2	L2	NoL2	L2	NoL2	L2
ResNet18	65.6 / 92.9	97.4 / 14.8	77.1 / 80.6	88.7 / 55.4	80.4 / 75.4	91.1 / 44.6	89.8	92.6

Table 3: Comparison of L2 vs NoL2 ResNet18 models, when both are trained for only 60 epochs. AUROC/FPR95 scores are shown, using feature norms as the scoring rule. L2 normalization substantially speeds up feature learning, and also improves OoD performance. After only 60 epochs, OoD performance is better than NoL2 models trained for the full 350 epochs, and accuracy is nearly as good (see Table 2).

	SVHN		**CIFAR100**		**TinyImageNet**	
Method	Norm	SM	Norm	SM	Norm	SM
VGG16	95.8 / 28.5	94.8 / 36.4	88.4 / 59.1	87.9 / 60.8	89.9 / 50.9	89.3 / 53.6
ResNet18	97.4 / 14.8	96.8 / 18.8	88.7 / 55.4	88.7 / 55.9	91.1 / 44.6	90.7 / 47.1
ResNet50	98.6 / 7.80	98.5 / 9.10	90.4 / 49.3	90.5 / 49.5	91.9 / 43.8	91.9 / 43.4

Table 4: Comparison of L2 models with AUROC/FPR95 scores, using feature norms and softmax as scoring rules. VGG16 and L2 ResNet18 are trained for 60 epochs, and ResNet50 is trained for 100 epochs. Scaled softmax performs comparably, but is slightly worse in general. Note that for the softmax case, norms must be scaled down to mitigate saturation, and this requires tuning the scaling parameter on test data.

	SVHN			CIFAR100		
Method	Norm - Feature	Norm - Logits	MaxLogit	Norm - Feature	Norm - Logits	MaxLogit
Min	96.3 / 6.93	95.9 / 7.51	95.6 / 9.12	88.2 / 52.1	88.5 / 51.0	88.3 / 52.6
Max	98.6 / 23.5	98.4 / 24.3	98.1 / 27.9	89.4 / 57.6	89.5 / 56.3	89.4 / 57.9
Mean	97.4 / 14.8	97.2 / 15.9	96.9 / 18.7	88.7 / 55.4	88.9 / 53.9	88.8 / 55.8

Table 5: Comparison of feature norm, logit norm and MaxLogit measurements over 15 ResNet18 L2 60 model seeds, for AUROC / FPR95 metrics. Using logit norms or MaxLogit provides no benefits over feature norms for near or far-OoD data.

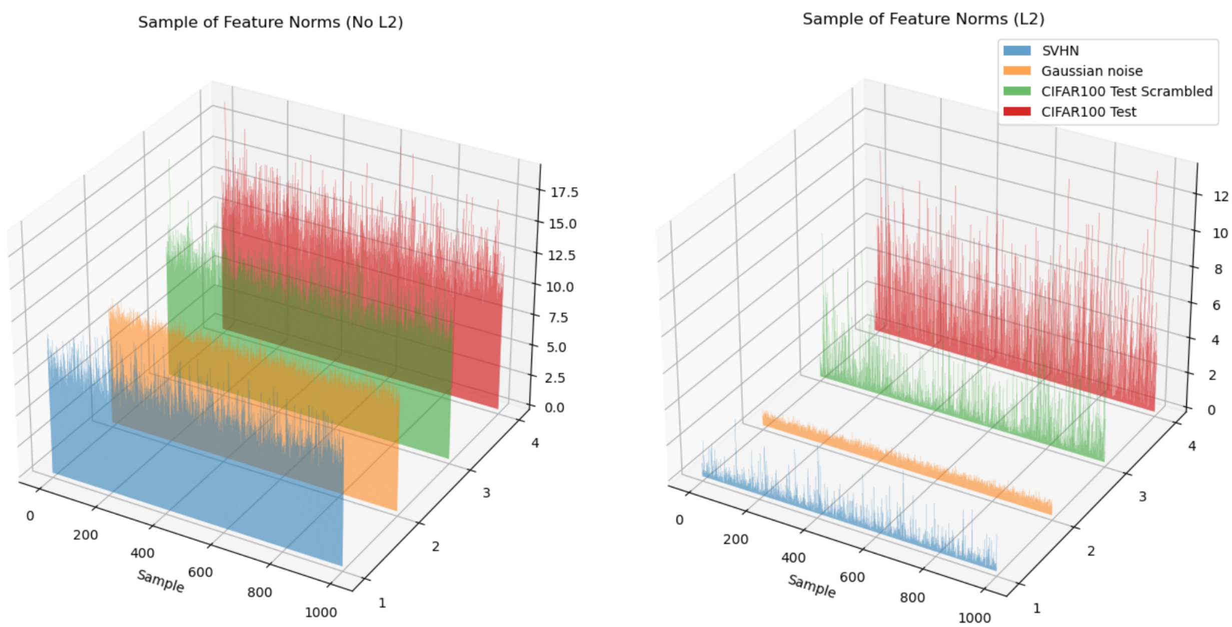

Figure 7: Comparison of norms under NoL2 ResNet50 350 (left) and L2 ResNet50 150 (right) models, four datasets (CIFAR100 Test, SVHN, Gaussian noise and pixel-wise scrambled CIFAR100 Test). See Section A.3 for discussion.

A.3 CIFAR100 Results

The behaviour of our method using CIFAR100 as a more complex ID dataset is as expected. A standard ResNet18 does not have enough capacity to perform reasonably well on CIFAR100, so we use ResNet50. The left side of Figure 7 shows that for the NoL2 case, input invariance behaviour is still present: completely different datasets produce very similar sized norms on average, and there is limited intra-dataset variance. As discussed in Section 3.2 and Figure 4, this is most likely due to the equinormality constraint inherent in CE loss. When analyzing these same datasets in the L2 case, inter-dataset sizes are substantially different, and intra-dataset variability increases as well. Performance is not as high as with CIFAR10, but this is standard in the literature, as CIFAR100 is a substantially more difficult task. Intuitively, a model that does not excel at ID data cannot be expected to excel at recognizing OoD data: accuracy on the CIFAR100 test set is only around 0.75 (Table 6).

	SVHN		CIFAR10		TinyImageNet		Accuracy	
Model	No L2	L2	No L2	L2	No L2	L2	No L2	L2
ResNet50	83.0 / 63.9	91.1 / 52.6	73.2 / 84.5	77.1 / 80.9	73.4 / 82.8	84.2 / 66.6	74.2	74.7

Table 6: AUROC/FPR95 results for CIFAR100 on NoL2 ResNet50 350 and L2 ResNet50 150 models.