Learning ODIN

Amir Jevnisek
amirjevn@mail.tau.ac.il
Shai Avidan
avidan@eng.tau.ac.il

School of Electrical Engineering
Tel-Aviv University
Tel-Aviv, Israel

Abstract

ODIN is a popular Out-Of-Distribution (OOD) detection algorithm. It is based on the observation that using temperature scaling and adding small perturbations to the input can separate the softmax score distributions between in- and out-of-distribution images, allowing for more effective detection. Instead of passively making this observation, we derive a new loss, termed Gradient Quotient (GQ) loss, that encourages this behaviour by the network. GQ can be used either to train a classification network from scratch, or fine-tune it. We show theoretically why GQ encourages the observation made by ODIN and evaluate GQ on a number of network architectures and datasets. Experiments show that we achieve SOTA on a large number of standard benchmarks.

1 Introduction

Deep Neural Networks (DNN) must handle Out-Of-Distribution (OOD) samples, if they are to operate in the real world. This is because the data distribution of samples during inference rarely matches that of the data during training.

A long line of research suggested various approaches to solving this problem. Perhaps the most natural one is to look at the *softmax* response and reject samples whose max softmax score is below some pre-defined threshold [8]. However, it was already shown that neural networks can produce arbitrarily high softmax confidence for inputs far away from the training data [**?**].

A long list of authors used various cues (either on the data or the network) to detect OOD samples. For example, ODIN [13] observed that after using temperature scaling in the softmax function and adding small controlled perturbations to inputs, the softmax score gap between in - and out-of-distribution examples is further enlarged. This makes it easier to distinguish between the two.

An alternative approach was suggested by Lee *et al.* [11] who assume that pre-trained features can be fitted well by a class-conditional Gaussian distribution and use Mahalanobis distance with respect to the closest class conditional distribution to determine if a test sample is normal or not. Similarly, Amersfoort *et al.* [22] use uncertainty in estimation by measuring the distance to the closest class centroid in feature space.

Recently, it was suggested that instead of looking at the softmax, one should consider an energy based score function [15], which was shown empirically to give results that are

superior to softmax-based scores. This was later extended by Lin *et al.* [14] who use a multi-level approach that relies on all layers of the network, and not just the last one.

A different direction was suggested by Serrà *et al.* [20]. They pose that OOD can be detected based on input complexity, and suggest using generative models to learn the likelihood of the data.

Common to these methods is that they make some observation about the data (or the network) and conduct their OOD detection after the fact. This makes sense, as they do not want to interfere with the actual training of the network. These cues are usually weak and an alternative approach is to use OOD exposure. For example, Papadopoulos *et al.* [19] add a regularization term to the loss function such that the model produces a uniform distribution for OOD samples. Yu and Aizawa [26] use unlabeled OOD data to maximize the discrepancy between the decision boundaries of two in-distribution classifiers to push OOD samples outside the manifold of the in-distribution (ID) samples.

Observe that this assumes that OOD samples are available during training. Shafaei *et al.* [21] conducted exhaustive tests and have shown that in realistic applications with high dimensional images this approach suffers from low accuracy and is not reliable in practice.

Turning our attention back to ODIN, we take a different approach. Instead of *assuming* that the ODIN observations hold, we *modify* the loss function of the underlying classifier to reflect them. That is, during training we encourage the softmax scores to be such that the controlled perturbations of the input will increase the softmax score gap between in- and out-ouf-distribution samples. This makes it easier for ODIN to detect OOD.

Specifically, we train the classification network with a new loss term that consists of the sum of the standard cross-entropy loss a new loss function, termed Gradient Quotient, that serves as a regularizer. We also show that GQ can be used to fine-tune a pre-trained network. At test time, we use the network with ODIN on top of it. If ODIN declares the test sample to be in-distribution, we return the classification result of the network. Otherwise, we report that the sample is OOD.

We thoroughly evaluate our approach on multiple network architectures and multiple datasets, and find that GQ does indeed help ODIN achieve SOTA results. We also conduct extensive experiments to quantify the tradeoff between improving OOD detection and maintaining in-distribution classification accuracy, and find that GQ usually increases OOD detection, as expected, at the cost of a slight degradation in in-distribution classification accuracy. To summarize:

- We introduce a new loss term, Gradient Quotient (GQ), that helps regularize cross-entropy loss for better OOD detection.
- We show analytically why GQ helps separate the softmax scores of in- vs. out-of-distribution samples
- We evaluate GQ on several different network architectures and several different datasets, achieving SOTA on many of them. Our code is available at: https://github.com/ajevnisek/learning-odin.

2 Gradients Quotient

ODIN, or Out-Of-Distribution Detection [12], makes two contributions. First, they use temperature scaling to help separate the softmax scores between in- and out-of-distribution im-

ages. Then, they observe that perturbing a query sample in the direction of the gradient can have stronger effect on the in-distribution images than that on out-of-distribution images.

Following [12], let $f = (f_1, ...f_N)$ denote a neural network trained to classify N classes. The network takes as input an In-Distribution sample x and computes softmax scores $S_i(x;T)$, controlled by temperature T:

$$S_i(x;T) = \frac{exp(f_i(x)/T)}{\sum_{j=1}^{N} exp(f_j(x)/T)}. \tag{1}$$

The maximum softmax probability $S_{\hat{y}}(x;T) = \max_i S_i(x;T)$ is termed the softmax score, and the label assigned by the network is taken to be: $\hat{y}(x) = \arg\max_i S_i(x)$.

Now, ODIN claims that for an In-Distribution sample x, with a perturbation:

$$\tilde{x} = x - \varepsilon sign(\nabla_x S_{\hat{y}}) \tag{2}$$

The gap: $S_{\hat{y}}(\tilde{x}) - S_{\hat{y}}(x)$ will be higher, on average, than the same gap for Out-of-Distribution samples. ODIN's decision rule is therefore:

$$|S_{\hat{y}}(\tilde{x}) - S_{\hat{y}}(x)| \underset{OOD}{\overset{ID}{\gtrless}} Th. \tag{3}$$

We suggest training a neural network such that this is a design property of the network's weights. We do that by adding a loss term to the training procedure of the neural network. To derive the loss term, take the Taylor expansion of the softmax with the perturbed image $S_{\hat{y}}(\tilde{x})$ around x:

$$S_{\hat{y}}(\tilde{x}) = S_{\hat{y}}(x) + \varepsilon \cdot \nabla_x S_{\hat{y}}(x) + \text{higher orders of } \varepsilon \tag{4}$$

Plugging this into the ODIN decision rule, we get:

$$|S_{\hat{y}}(\tilde{x}) - S_{\hat{y}}(x)| = |S_{\hat{y}}(x) + \varepsilon \cdot \nabla_x S_{\hat{y}}(x) - S_{\hat{y}}(x)| = |\varepsilon \cdot \nabla_x S_{\hat{y}}(x)| \gtrless Th. \tag{5}$$

Equation 5 implies that under a first order Taylor expansion, ODIN's decision rule is equivalent to requiring a large softmax gradient (i.e., a large $\nabla_x S_{\hat{y}}(x)$). The term 'large' is ill-defined ('large' compared to what?). Instead, we introduce a *Gradient Quotient* (GQ) loss term:

$$\mathcal{L}_{GQ} = \frac{||\nabla_x S_{\hat{y}}||}{\sum_{y_j \neq \hat{y}} ||\nabla_x S_{y_j}||} \tag{6}$$

where y is the true label, $\hat{y}$ is the estimated class label, y_j are all class label which are not the true class label y. This way, the term 'large' refers to the softmax scores of the rest of the class labels. The loss encourages the softmax of the correct label to be higher than that of the rest of the classes. Observe that we can encourage the ODIN assumption during training because we have access to the ground truth label y, and we only use in-distribution samples. The full loss term we train our networks with is:

$$L = L_{CE}(y, \hat{y}) + \lambda \cdot \frac{||\nabla_x S_y||_1}{\sum_{y_j \neq y} ||\nabla_x S_{y_j}||_1} \tag{7}$$

where L_{CE} is the standard cross-entropy loss.

Our full algorithm is given in Algorithm 1. We train the network using the loss function in Equation 7. At inference time, we use the network and on top of it run ODIN. If ODIN determines the test sample is in-distribution we return the result produced by the network, else we report that the test sample is OOD.

Algorithm 1 Learning ODIN

Inputs (1) In-Distribution dataset $\mathcal{D}_{ID}^{train}$, (2) classifier $f_\theta(\cdot)$ with parameters θ.
Outputs: (1) An OOD detector, (2) An In-Distribution classifier.

1: Train $f_\theta(\cdot)$ with the following loss: $L = L_{CE} + \lambda \cdot \mathcal{L}_{GQ}$
2: Create $g(x)$ an ODIN OOD-detector on f_θ's logits.
return $g(x)$ is the OOD-detector, f_θ is the In-Distribution classifier

We first evaluate the impact of GQ on the separation of softmax scores between in- and out-of-distribution samples. Figure 1 shows the distribution of the Log-Softmax Gradient w.r.t Image-Pixels :$||\nabla_x \log S_{\hat{y}}(x;T)||_1$). In this case, the base network was trained on CIFAR-10, and the dashed black line shows the density of Log-Softmax Gradients it produces. The colored lines show the density of Log-Softmax Gradients for various OOD datasets. There is indeed a nice separation between the in-distribution samples and the out-of-distribution samples, as observed by ODIN. But, when training the network using GQ, the separation increases dramatically, as shown by the solid black line. This suggests that the separation is much larger now, making it easier for ODIN to detect OOD samples.

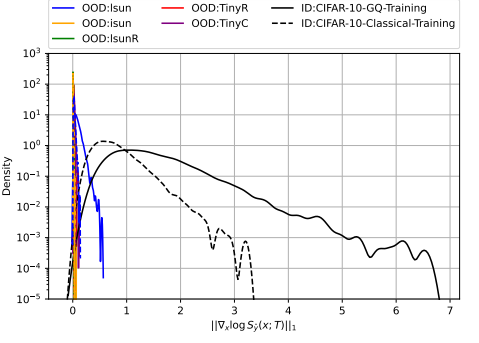
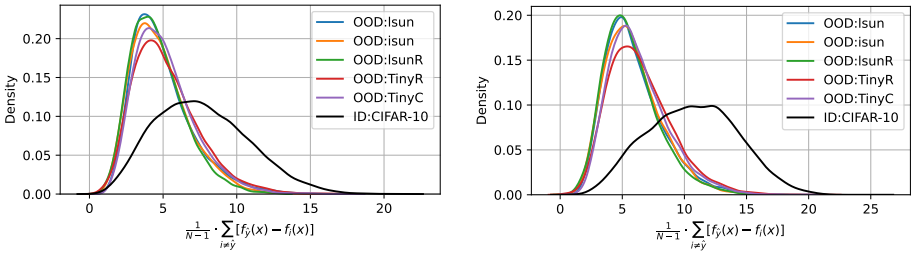


Figure 1: **Log-Softmax Gradient w.r.t Image-Pixels Distribution** $||\nabla_x \log S_{\hat{y}}(x;T)||_1$. When our loss term is added, the distribution for In-Distribution samples is wider than in the case of classical training (CE-loss). We refer the reader to a zoom-in of this figure in the supplemental to witness that all OOD datasets but one did not have significant support growth.

Following ODIN, we next look at the statistics of the deviation of the network's maximal logit from all other logits, where we define *logits gap* to be the average deviation of the maximal logit from all other logits. Figure 2(a) shows the logits gap for a model that was trained solely with Cross-Entropy (CE) loss. Figure 2(b) shows the logits gap for a model that was trained with our Gradient Quotient (GQ) loss regularization. It is clear that the distribution of the logit gap for In-Distribution samples shifted to the right when the model is trained with GQ compared to when it was trained using only CE-loss. Furthermore, one can observe that for OOD data, the distribution did not shift. This validates the observation that neural networks tend to make more confident predictions on in-distribution images. This observation was made in ODIN [12], and encoded explicitly in the loss function of [19] using OOD data. We encode it into the loss function *without* using OOD data.



(a) (b)

Figure 2: **Logit Gap Distribution** *logits gap* is defined to be the average deviation of the maximal logit from all other logits. (a) shows the *logits gap* for a model that was trained using cross-entropy loss. (b) shows the *logits gap* for a model which was trained with our method.

3 Experimental Results

3.1 Setup

In-Distribution Datasets We take the standard course of CIFAR-10 and CIFAR-100 [9] as In-Distribution datasets. We use the standard split, with $50K$ train images and $10K$ test images. For higher resolution images we take Imagenet-30 as an In-Distribution dataset. In the latter case, we resize images to 254×254.

Out-of-Distribution Datasets We use OOD datasets only at test time. We do not train or fine-tune using OOD datasets. We followed [14] and evaluated the performance of OOD detection on a total of 10 datasets: MNIST [10], K-MNIST [3], fashion-MNIST [23], LSUN (crop) [25], SVHN [18], Textures [2], STL10 [4], Places365 [27], iSUN [24] and LSUN (resize) [25]. When CIFAR-10 and CIFAR-100 images are used as In-Distribution, we resize all images to 32×32 pixels. As for the case of Imagenet-30 as OOD, we resize all images to 254×254. For each OOD dataset, we evaluate on the entire test split.

Evaluation Metrics We evaluate our method with three metrics (1) Area under the Receiver Operating Characteristic curve (AUROC), (2) False Positive Rate (FPR95) on OOD data when the true positive rate for in-distribution data is 95%, and (3) In-Distribution test-accuracy, that is the accuracy of the classifier on the in-distribution test set. We follow [14, 15] and report average AUROC scores over the datasets under test and average FPR95 over the test datasets.

Architectures We evaluate our approach on a number of common architectures. The first is a simplified ResNet [8] architecture which was suggested by [17], which we term Madry's ResNet. Next, we evaluate our approach on two native ResNet architectures: ResNet18 and ResNet34. See supplemental for details about the number of parameters of the different architectures discussed in this paper.

Training Details We train all models using SGD with learning rate starting at 0.1 and controlled by a Cosine Annealing scheduler [16]. The scheduler's maximal number of iterations T_{max} is set to the number of epochs. We train all models for 200 epochs, the batch size is 64 and the weight λ controlling the balance between CE-loss and our regularization loss term is set to 10^{-6}. We follow [14] and use ODIN and Mahalanobis perturbation amplitude $\varepsilon = 0.001$ for CIFAR-10 and ImageNet-30 and $\varepsilon = 0.006$ for CIFAR-100 as In-Distribution datasets. The temperature used to scale softmax values as in ODIN is set to $T = 1000$.

Architecture	Method	AUROC ↑ (%)	FPR95 ↓ (%)	ID-Accuracy ↑ (%)
WideResNet-40-4	MSP	88.99	56.81	94.93
	ODIN	90.11	35.31	94.93
	Mahalanobis	89.33	35.48	94.93
	Energy	90.04	35.26	94.93
MSDNet Exit@last	MSP	89.72	49.87	94.09
	ODIN	90.33	29.30	94.09
	Mahalanobis	82.84	75.19	94.09
	Energy	90.48	33.62	94.09
MSDNet (dynamic exit)	MOOD	91.26	28.05	94.13
Madry's Resnet /+ GQ	ODIN	90.51 / 91.94	36.57 / 30.19	92.19 / 93.29
ResNet18 /+ GQ	ODIN	89.09 / **93.55**	27.75 / **25.14**	95.27 / 94.26
ResNet34 /+ GQ	ODIN	89.03 / 89.08	35.33 / 27.63	95.21 / **95.56**

Table 1: **OOD detection performance: CIFAR-10 as In-Distribution:** OOD detection performance comparison between our method (termed GQ) and baseline methods: MSP [7], ODIN [12], Mahalanobis [11], Energy [15] and MOOD [14]. All results are averaged across 10 datasets. AUROC is the Area Under the Receiver Operating Curve, FPR95 is the False Positive Rate when the True Positive Rate (classifying ID samples as ID samples) is at 95%, and ID-Accuracy is the classifier's accuracy for In-Distribution samples. Best results in each column are marked in **Bold** and second best in underline. See supplementary for detailed results for each OOD test dataset. For the networks we trained, classical training with CE-loss is separated from CE-loss+GQ with a '/'. As can be seen, adding GQ to any of the architectures improves results and our approach (GQ) achieves best results in all measures compared to all baseline models.

3.2 Results

Does GQ Help OOD Detection? Tables 1 and 2 summarize our results compared to common Out-of-Distribution benchmarks. The tables show average performance metrics across 10 different OOD datasets for CIFAR-10 and CIFAR-100 as In-Distribution datasets. We show the full evaluation details for all 10 OOD datasets in the Supplemental. Our approach uses only in-distribution data and does not require any auxiliary OOD data. Therefore, the reference methods and architectures we consider are those that do not require OOD samples (or OOD exposure) at train time. In particular, we compare against the following baseline methods: MSP [7], ODIN [12], Mahalanobis [11], Energy [15] and MOOD [14].

Does GQ Hurt In-Distribution Classification? Focusing on the CIFAR-10 case (Table 1), we observe that adding GQ either improves AUROC by more than 4% (in the case of ResNet18), at the cost of degrading in-distribution accuracy by 1%. In the case of Madry's Resnet, both AUROC and in-distribution accuracy improves by more than 1%. And in the case of ResNet34 we observe no impact of GQ. When considering CIFAR-100 (Table 2), we observe that adding GQ does hurt AUROC in the case of ResNet34, but in the rest of the cases has a limited impact.

Fine Tuning and High Resolution Images: Experiments so far focused on images of size 32×32 pixels. We next evaluate GQ on high resolution images. To do that, we follow [1] and use ResNet-18 and ResNet-101 models pretrained on ImageNet. Instead of training these networks from scratch, we fine-tune the pre-trained model solely with the $\mathcal{L}_{GQ}$ loss term for one epoch and use that as our network. As an In-Distribution dataset we take ImageNet-30,

Architecture	Method	AUROC ↑ (%)	FPR95 ↓ (%)	ID-Accuracy ↑ (%)
WideResNet-40-4	MSP	77.10	77.51	76.90
	ODIN	84.66	57.22	76.90
	Mahalanobis	83.19	**53.52**	76.90
	Energy	83.69	62.71	76.90
MSDNet Exit@last	MSP	78.33	76.71	75.43
	ODIN	84.89	57.45	75.43
	Mahalanobis	73.80	78.06	75.43
	Energy	84.51	59.15	75.43
MSDNet (dynamic exit)	MOOD	84.97	75.22	75.26
Madry's Resnet /+ GQ	ODIN	83.69 / 83.46	55.81/ 60.13	70.93 / 71.17
ResNet18 /+ GQ	ODIN	85.95 / **86.88**	63.42 / 60.31	**78.40** / 77.72
ResNet34 /+ GQ	ODIN	84.46 / 81.98	59.81 / 58.23	**78.40** / 78.27

Table 2: **OOD detection performance: CIFAR-100 as In-Distribution:** OOD detection performance comparison between our method (termed GQ) and baseline methods: MSP [7], ODIN [12], Mahalanobis [11], Energy [15] and MOOD [14]. All results are averaged across 10 datasets. AUROC is the Area Under the Receiver Operating Curve, FPR95 is the False Positive Rate when the True Positive Rate (classifying ID samples as ID samples) is at 95%, and ID-Accuracy is the classifier's accuracy for In-Distribution samples. Best results in each column are marked in **Bold** and second best in underline. See supplementary for detailed results for each OOD test dataset. For the networks we trained, classical training with CE-loss is separated from CE-loss+GQ with a '/'.

and resize all images in this experiment to be 254×254 pixels. We compare our results to that of [1]. In addition, we run pNML on top of our fine-tuned network. Results for ResNet-18 are summarized in table 3. ResNet-101 results are found in the Supplementary Material.

We observe that GQ outperforms the baseline method (both for ResNet-18 and ResNet-101) in all cases. Further, observe that in the case of the saturated OOD datasets, GQ is on par with pNML and that running pNML on top of GQ does not contribute much to the AUROC detection results. On the other hand, in the case of ResNet-18 as a backbone, while pNML struggles with detecting CIFAR-100 as OOD, we cut the error by 50%: from 92.15% AUROC to 98.55% AUROC. Moreover, while the classification accuracy for Imagenet-30 for the pre-trained ResNet18 on Imagenet is 69.57%, one epoch of fine-tuning with $\mathcal{L}_{GQ}$ increases the classification accuracy to 80.87%.

This experiment demonstrate that our method can work with high resolution images, as well as low resolution ones. In addition, it shows that it is possible to use GQ for fine tuning, which lets us work with pre-trained models.

Is GQ sensitive to the tuning parameter λ? We validate the in-sensitivity of GQ to λ. We experiment six *lambda* decade values from 10^{-6} to 10^{-1}. AUROC and FPR95 metrics stay stable - under 0.5% peak-to-peak for AUROC and $\sim 2.5\%$ in FPR95. The latter metric is inherently noisier since it is induced from a single point in the ROC curve. In-Distribution accuracy falls by 0.7% for λs up to 10^{-1}. We refer the reader to the Supplemental for the full table and evaluation details.

Training Method		Baseline	pNML	GQ	GQ+pNML
ID-Accuracy		69.57	80.87	80.87	80.87
OOD	iSUN	95.58	99.74	99.53	99.47
	LSUN (R)	95.51	99.72	99.63	99.47
	LSUN (C)	96.89	99.77	99.77	98.92
	Uniform	99.35	99.99	99.46	100
	Gaussian	98.78	100	99.47	100
	SVHN	99.18	99.99	99.99	99.90
	CIFAR-10	89.99	99.79	99.85	99.20
	CIFAR-100	92.15	92.15	99.63	99.08

Table 3: **High Resolution Images OOD detection**. OOD detection for the case of Imagenet-30 as In-Distribution dataset and ResNet-18 as a backbone. Baseline is the case of the pre-trained backbone. We cite AUROC results for the case of pNML from [1], we evaluate AUROC for the $\mathcal{L}_{GQ}$ fine-tuned networks and term it GQ, we apply pNML on top of the fine-tuned network and denote it as GQ+pNML.

4 Related Work

ODIN [12] introduced sample perturbation as a way to detect OOD. Two other approaches that follow this idea are Energy based models [15] and pNML [1] that we discuss next.

Energy Based Models Vs. GQ: [15] use the energy score of a sample as an inverse score for the probability of the sample to be In-Distribution. They show how the derivative of the Negative-Log-Likelihood (NLL) encourages the energy of the correct class label to go down and the energy of all other labels to go up. Recall that for a classifier with logits $f_y(x)$, the energy of a sample (x,y) is $E(x,y) = -f_y(x)$. The NLL with temperature in terms of the energy is:

$$\mathcal{L}_{nll} = \mathbb{E}_{(x,y)\sim P^{in}}[-\log \frac{e^{f_y(x)/T}}{\sum_{j=1}^{N} e^{f_j(x)/T}}] = \mathbb{E}_{(x,y)\sim P^{in}}[\frac{1}{T}\cdot E(x,y) + log \sum_{j=1}^{N} e^{-E(x,j)/T}] \tag{8}$$

where P^{in} is the distribution of the training data (observe the similarity to equation 1). The derivative of the loss with respect to the network parameters is:

$$\frac{\partial \mathcal{L}_{nll}(x,y;\theta)}{\partial \theta} = \frac{1}{T}\frac{\partial E(x,y)}{\partial \theta}\cdot(1 - p(Y=y|x)) - \frac{1}{T}\sum_{j\neq y}\frac{\partial E(x,j)}{\partial \theta}\cdot p(Y=j|x) \tag{9}$$

Since the sign of the gradient is positive for $E(x,y)$ and negative for all $j \neq y$ the overall gradient encourages low $E(x,y)$ and high values for $E(x,j)$. [15] observed that the gradient of NLL w.r.t the *network parameters* pushes the energy of an ID sample downwards and pulls up the energy of all other labels. Similarly to ODIN, [15] makes an observation about the scores of the network, but do not change the way they are computed.

We, on the other hand, encourage the network to have peaky softmax. We push the *gradient* of the ground truth softmax to be larger than all other gradients. This gradient is calculated with respect to *image space pixels* (and not network parameters, as is done in [15]). If we assume softmax to correspond to the energy - we want the energy of the ground truth softmax to decay fast with changes in the input image. Faster than what? Faster than it decays when observing the energy with respect to classes which are not the ground truth.

	CIFAR-10		CIFAR-100	
OOD	pNML	GQ	pNML	GQ
iSUN	**97.5**	97.3	87.6	**87.9**
SVHN	**97.9**	96.8	**95.4**	85.0
LSUN-C	95.6	**98.5**	88.1	**94.6**
ImageNet-C	96.2	**98.2**	88.6	**92.6**
ImageNet-R	**96.6**	96.4	**88.5**	87.1
LSUN-R	97.7	**99.0**	88.0	**89.6**

Table 4: **pNML vs GQ (our) method:** Comparison with pNML [1]. We show results for CIFAR-10 (left columns) or CIFAR-100 (right columns) as the In-Distribution (ID) datasets. Rows show the different OOD datasets we evaluated against. **Bold** numbers indicate best results for a given ID and OOD dataset. Results of both methods are comparable.

pNML Vs. GQ: pNML, or predictive normalized maximum likelihood (pNML), was recently suggested by [1] for OOD detection. They show that the generalization error of pNML, denoted as the regret, can be used to detect OOD samples. Specifically, they work on the last layer of the Neural Network and do the following: Given a test sample, they add it to the training set with an arbitrary label, find the best-suited model, and take the probability it gives to the assumed label. They repeat this procedure for every label and normalize to get a valid probability assignment. Then they use the log normalization factor to compute the regret, which is used as a confidence measure. Large regret means that the test sample is more likely to be OOD. A nice feature of the pNML is that it does not make any assumption about the distribution of the data (either during train or test).

We compare our method to the ODIN+pNML method suggested in [1]. Specifically, they use a standard classification network (that is based on the DenseNet-BC-100 backbone), followed by ODIN. Then, they take the perturbed sample after running the ODIN procedure and use pNML to determine if it is OOD or not. Results are reported in table 4. As can be seen, there is no clear winner method - AUROC scores are mixed [1].

pNML also uses perturbation of the test sample to detect if it is OOD, just like we do. However, there are some major differences between the two approaches. We add a regularization term (Gradient Quotient) to the loss function. In contrast, pNML is built on the *individual setting*. For this setting, there is no assumption about how the training and the test data are generated, nor about their probabilistic relationship. Therefore, pNML serves as a complementary tool given a (trained) classifier. We, on the other hand, suggest a substitute for the loss function.

Adversarial Learning The goal of untargeted adversarial attacks is to generate samples such that they are misclassified. For targeted adversarial attacks, the misclassification is guided to a specific (target) class. FGSM [5] and PGD [17] attacks are based on perturbations made using the sign of the elements of the gradient of the cost function with respect to the input. Our loss term, GQ, encourages the gap of softmax outputs to be large under input perturbation. For adversarial attacks, this is a convenient platform. For targeted attacks and a CE-loss term, one can derive the following corollary:

$$\max_{\delta\in\Delta}\left(\ell(h_\theta(x+\delta),y)-\ell(h_\theta(x+\delta),y_{\text{target}})\right)\equiv\max_{\delta\in\Delta}\left(h_\theta(x+\delta)_{y_{\text{target}}}-h_\theta(x+\delta)_y\right) \quad (10)$$

where: ℓ is the CE-loss, $h_\theta(x)$ is the logit vector for an input image x, the correct class

[1] The models in table 4 are evaluated using pNML code [1].

label for this image is y, y_{target} is the target lable, δ is the input perturbation and Δ is the set of allowable input perturbations. One interpretation of Equation 10 is that a targeted attack maximizes the logit gap between the target class and the correct class. Our loss term indirectly supports this through maximization of the logit gap for the correct class. Figure 2 demonstrates that our loss term can encourage the minimization of $h_\theta(x+\delta)_y$. The latter is one ingredient in the maximization of the the left hand side of equation 10. We remark that a comprehensive study of the tie between adversarial learning and GQ is an interesting direction for future research.

5 Conclusion

We take an observation made in the past about the behaviour of OOD samples and turns it into a regularization term that can be used during training. This is done by introducing a new loss function, termed Gradient Quotient (GQ), that encourages the network to calculate softmax values that behave as expected by ODIN. This loss can be used either to train a network from scratch, or just fine-tune it. We have shown theoretically why GQ encourages the network to follow the observation made by ODIN and evaluate it extensively on a large number of datasets and network architectures.

This work was partly funded by ISF grant 1549/19.

References

[1] Koby Bibas, Meir Feder, and Tal Hassner. Single layer predictive normalized maximum likelihood for out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34, 2021.

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[3] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

[4] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[8] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.

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

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] 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.

[12] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[13] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1VGkIxRZ.

[14] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. Mood: Multi-level out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15313–15323, 2021.

[15] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020.

[16] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[18] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[19] Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 441:138–150, 2021.

[20] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2020.

[21] Alireza Shafaei, Mark Schmidt, and James Little. A Less Biased Evaluation of Out-of-distribution Sample Detectors. In *BMVC*, 2019.

[22] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. 2020.

[23] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[24] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

[25] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[26] Qing Yu and Kiyoharu Aizawa. Unsupervised out-of-distribution detection by maximum classifier discrepancy. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9517–9525. IEEE, 2019. doi: 10.1109/ICCV.2019.00961. URL https://doi.org/10.1109/ICCV.2019.00961.

[27] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.