Gradient-Regularized Out-of-Distribution Detection

Sina Sharifi[1*], Taha Entesari[1*], Bardia Safaei[1*],
Vishal M. Patel[1], and Mahyar Fazlyab[1]

Johns Hopkins University, Baltimore MD 21218, USA
{sshari12, tentesa1, bsafaei1, vpatel36, mahyarfazlyab}@jhu.edu

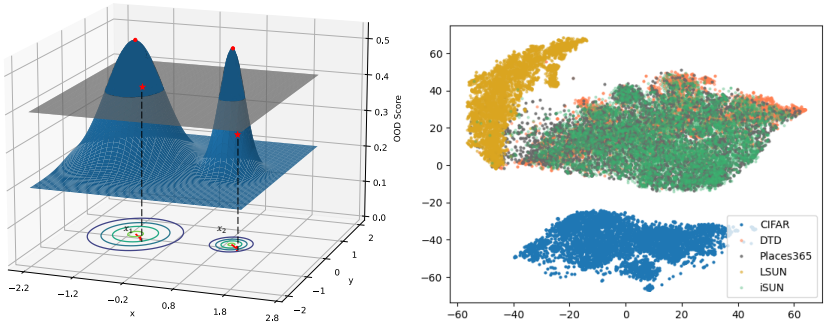
Abstract. One of the challenges for neural networks in real-life applications is the overconfident errors these models make when the data is not from the original training distribution. Addressing this issue is known as Out-of-Distribution (OOD) detection. Many state-of-the-art OOD methods employ an auxiliary dataset as a surrogate for OOD data during training to achieve improved performance. However, these methods fail to fully exploit the local information embedded in the auxiliary dataset. In this work, we propose the idea of leveraging the information embedded in the gradient of the loss function during training to enable the network to not only learn a desired OOD score for each sample but also to exhibit similar behavior in a local neighborhood around each sample. We also develop a novel energy-based sampling method to allow the network to be exposed to more informative OOD samples during the training phase. This is especially important when the auxiliary dataset is large. We demonstrate the effectiveness of our method through extensive experiments on several OOD benchmarks, improving the existing state-of-the-art FPR95 by 4% on our ImageNet experiment. We further provide a theoretical analysis through the lens of certified robustness and Lipschitz analysis to showcase the theoretical foundation of our work. We will publicly release our code after the review process.

Keywords: Out-of-Distribution Detection, Gradient Regularization, Energy-based Sampling

1 Introduction

Neural networks are increasingly being utilized across a wide range of applications and fields, achieving unprecedented levels of performance and surpassing traditional state-of-the-art approaches. However, concerns about robustness and safety, coupled with significant challenges in verifying robustness and ensuring safe performance impede their use in more sensitive applications. One concern that arises when deep models are deployed in the real world is their tendency to produce over-confident predictions upon encountering unfamiliar samples that are far distant from the space in which they were trained [39]. As a result, the

[*]Equal contribution



(a) (b)

Fig. 1: *Left:* Effect of the local structure of the score manifold on a two-dimensional toy example for OOD detection. The grey plane depicts the score function's decision threshold. An equal amount of perturbation in these two scenarios results in different OOD detections, highlighting the importance of the local structure of the score manifold. *Right:* t-SNE plot showing the representation of ID and OOD datasets for CIFAR experiments.

field of Out-of-Distribution (OOD) detection has emerged to study and address this phenomenon.

Many works have been presented in recent years, trying different methods and ideas to regulate the network's output to OOD samples in terms of softmax probability [17, 18] or energy metrics [30, 35]. One family of methods, known as post-hoc approaches [1, 17, 28, 31, 47], operate on a given trained network and employ certain statistics, such as the confidence level of the network, the number of active neurons, the contribution of important pathways in the network, etc., to perform OOD detection without using any samples from an auxiliary distribution. Other methods [6, 18, 22, 30, 35] use OOD data for training or finetuning a network. The latter family usually tends to achieve better performances in comparison to post-hoc methods, as their access to the auxiliary dataset provides them with additional information regarding the outlier distribution.

In this paper, we propose *GReg*, a novel **G**radient **Reg**ularized OOD detection method, which aims to learn the local information of the score function to improve the OOD detection performance. Using information from the gradient of the score function, our method builds on top of current state-of-the-art OOD detection methods (that require training) to incorporate local information into the training process and lean the network towards a smoother manifold.

To motivate this idea, consider Figure 1a. This figure shows a two-dimensional example of a possible scenario in OOD detection. In this example, x_1 and x_2 are two OOD samples and the network has learned to assign the same OOD scores (denoted by red dots in the figure) to both samples and detect them correctly as OOD. However, the difference in the local behavior of the network at these points, leaves samples in the vicinity of x_2 susceptible to misdetection, whereas, samples close to x_1 are always detected as OOD. To see this, consider the per-

turbations indicated by the red arrows, where the resulting points in each case represent two possible samples with equal distance from points x_1 and x_2, respectively. However, as can be seen in the figure, the OOD detection algorithm cannot detect one of the samples correctly. This emphasizes the importance of regulating the local behavior of deep learning models, especially when OOD robustness is crucial. Furthermore, Figure 1b shows the t-SNE plot of the CIFAR and some of the well-known OOD benchmarks. This further confirms that even in high dimensions the ID data (CIFAR) are well-separated from the OOD data.

On top of that, several recent works have shown the effectiveness of some form of informative sampling or clustering on OOD detection [6, 22, 35]. Sampling gains more importance specifically when a large dataset of outlier samples is available and the network cannot be exposed to all of the data during training. Inspired by such works, we present *GReg+*, the coupling of *GReg* with an energy-based clustering method aimed at making better use of the auxiliary data during training. This is in line with our intuition of utilizing local information as clustering enables the use of samples from diverse regions of the feature space. The diversity put forward by the clustering mechanism allows *GReg+* to achieve its state-of-the-art performance in more regions of the space rather than only performing well on regions well-represented in the dataset.
In summary, our key contributions are:

- We propose the idea of regularizing the gradient of the OOD score function during the training (or fine-tuning). This allows the network to better learn the local information embedded in ID and OOD samples.
- We propose a novel energy-based sampling method that chooses samples –based on their energy levels– from the OOD dataset that represent more vulnerable regions of the space using clustering techniques.
- We provide empirical results and ablation studies on a wide range of architectures and different datasets to showcase the effectiveness of our method, along with an extensive comparison to the state-of-the-art. In addition, we provide a detailed theoretical analysis to justify our results.

To the best of our knowledge, this is the first work that utilizes the norm of the gradient of the score function during the training phase to learn the local behavior of the ID/OOD data aiming to improve the OOD detection performance.

2 Related Work

Several works have attempted to enable deep neural networks with OOD detection capabilities in recent years [2, 5, 16, 26, 31, 34, 40, 41, 43, 50]. These works can be divided into two main groups: methods that do not use auxiliary data (post-hoc) and those that utilize auxiliary data.

2.1 Post-hoc OOD Detection

In one of the earliest attempts to remedy the OOD detection problem, [17] uses the maximum softmax probability (MSP) score to recognize OOD samples

during inference. ODIN [18] improves the MSP score's separability by adding small noise to the input and utilizing temperature scaling. [20, 29–31] aim to formulate enhanced scoring functions to better distinguish OOD samples from ID samples. For example in [30], the authors propose using energy as a powerful OOD score. GradNorm [20] uses the vector norm of the gradient of the KL divergence between the softmax output and a uniform probability distribution to devise an OOD score. It was later observed by [46], that OOD data tend to have different activation patterns. As a solution, they propose activation truncation to improve OOD detection. Building on this observation, DICE [47] ranks weights based on the measure of contribution, and selectively uses the most contributing weights. LINe [1] further improves on DICE by using Shapley values [44] to more accurately detect important and contributing neurons.

2.2 OOD Detection with Auxiliary Outlier Dataset

When an auxiliary dataset is available, this group of methods [10,17,30] use this data to train the model to improve the network's robustness against OOD data. OE [17] and Energy [30] propose their respective loss functions to be used during the training of the network. Recently, [7] improved [30] by proposing balanced Energy loss that balances the number of samples of the auxiliary OOD data across classes. Furthermore, [42] increases the expected Frobenius Jacobian norm difference between ID and OOD. This is in sharp contrast with our intuition as we aim to decrease the norm of the score function in all regions. Another line of work [11,14,24,36,37] focuses on generating artificial samples that resemble the real OOD data, using various generative models such as GANs. Most recently OpenMix [55] was proposed as a Misclassification detection method, teaching the model to reject pseudo-samples generated from the available outlier samples.

Outlier Sampling: When large amounts of auxiliary data are available, choosing an informative and diverse set of samples to perform training becomes a priority. Many of the state-of-the-art OOD methods use their custom sampling techniques. NTOM [6] uses greedy outlier sampling where outliers are selected based on the estimated confidence. POEM [35] proposes a posterior sampling-based outlier mining framework for OOD detection. Most recently DOS [22] used K-Means [32] to perform clustering on the auxiliary dataset to provide the network with diverse informative OOD samples.

3 Preliminaries

3.1 Notation

We consider a classification setup where $\mathcal{X} \in \mathbb{R}^d$ denotes the input space and $\mathcal{Y} \in \{1, 2, ..., K\}$ denotes the labels. We define a neural network $f : \mathbb{R}^d \to \mathbb{R}^K$ where $f_i(x)$ denotes the i-th logit of the neural network to an input x. We denote the feature extractor of the neural network as h, which is followed by a classification layer. We use $f(x) = Wh(x) + b$ for all the models in this paper.

We denote the norm of a vector x as $\|x\|$. The log-sum-exp function is defined as $\text{LSE}(x) = \log \sum_{i=1}^{n} \exp(x_i)$. For simplicity, for a given score function S, we use $S(x)$ as a shorthand for $S(f(x))$.

The In-Distribution (ID), Out-of-Distribution (OOD) and Auxiliary distribution over $\mathcal{X}$ are denoted by $\mathcal{D}_{\text{in}}$, $\mathcal{D}_{\text{out}}$ and $\mathcal{D}_{\text{aux}}$, respectively. The corresponding datasets are assumed to be sampled i.i.d from their respective distributions. We denote the indicator function of an event $e \leq 0$ as $\mathbb{I}_{e\leq 0}$, i.e., if $e \leq 0$ then $\mathbb{I}_{e\leq 0} = 1$, and otherwise, $\mathbb{I}_{e\leq 0} = 0$. We denote $\mathcal{B}(x, \varepsilon) = \{y | \|y - x\| \leq \varepsilon\}$. Finally, a function f is said to be locally Lipschitz continuous on $\mathcal{X}$ if there exists a positive constant L such that

$$\|f(x) - f(y)\| \leq L\|x - y\|, \qquad \forall x, y \in \mathcal{X}.$$

The smallest such constant is called the local Lipschitz constant of f.

3.2 Out-of-Distribution Detection

It has been observed that when neural networks are faced with samples from a different distribution compared to their training distribution, they produce overconfident errors [16]. The goal of OOD detection is to differentiate the ID data from the OOD data. To achieve this goal, the main approach in the literature is to define a scoring function S and use a threshold γ to distinguish different samples. That is, for a sample x, if $S(x) \leq \gamma$, we label it as ID, and label it as OOD otherwise.

Most relevant to our setup, [30] uses the scoring function $S_{\text{En}}(x) = -\text{LSE}(f(x))$, and defines the following loss function to train the network to better differentiate the ID and OOD samples,

$$\begin{aligned}\mathcal{L}_{S_{\text{En}}} = {} & \mathbb{E}_{(x_{\text{in}}, y_{\text{in}})\sim\mathcal{D}_{\text{in}}}[\mathbb{I}_{S_{\text{En}}(x_{\text{in}})\geq m_{\text{in}}}(S_{\text{En}}(x_{\text{in}}) - m_{\text{in}})^2] \\ & + \mathbb{E}_{(x_{\text{aux}}, y_{\text{aux}})\sim\mathcal{D}_{\text{aux}}}[\mathbb{I}_{S_{\text{En}}(x_{\text{aux}})\leq m_{\text{aux}}}(m_{\text{aux}} - S_{\text{En}}(x_{\text{aux}}))^2],\end{aligned} \tag{1}$$

where m_{in} and m_{aux} define two thresholds to filter out points that already have an acceptable level of energy, i.e., if the energy score of an ID (OOD) sample is low (high) enough, it will be excluded from the energy loss.

4 Method

In this section, we propose our method which consists of regularizing the gradient of the score function by adding a new term to the loss, coupled with a novel energy-based sampling method to allow us to choose more informative samples during training. Figure 2 provides an overview of our method *GReg+* with a simple illustration of how the sampling algorithm utilizes clustering to provide more informative samples.

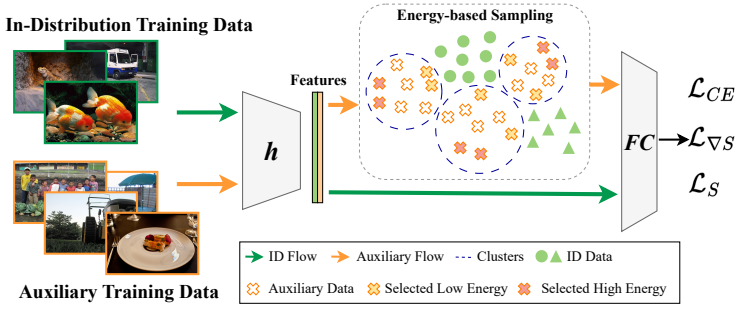


Fig. 2: Overview of *GReg+*. The gradient loss allows the method to obtain more local information from the training data. The sampling algorithm uses the normalized features of the OOD samples to perform clustering and choose the sampled data based on the energy score (see Algorithm 1) which will be used to calculate the gradient loss to expose the network to samples that can improve the performance of the model.

4.1 Gradient Regularization

Many of the current state-of-the-art methods focus only on the value of their scoring function S and aim at minimizing or maximizing it on different regions of the input space [7, 18, 22, 30, 35, 55]. We argue that by optimizing only based on the value of the score function, we are not utilizing all of the information that is embedded in the data. As such, the generalization of the model requires the auxiliary dataset to be a good representation of the actual OOD distribution and its corresponding support in the image space (which is extremely high-dimensional). In response, we aim to use the local information embedded in the data and propose to do so by leveraging the gradient of the score function S. In other words, we focus not only on the value of the score function S on the auxiliary samples but also on the local behavior of the score function around these data points. The rationale behind this approach is that, based on the definition of the problem, the distribution of ID and the OOD tends to be well separated. Therefore, the area close to an ID sample most likely does not include any OOD samples, and vice versa. Driven by this insight, we propose to add a regularization term $\mathcal{L}_{\nabla S}$ that promotes the smoothness of the score function around the training samples by penalizing the norm of its gradient.

Suppose x is a training sample. Using the first-order Taylor approximation of the score function around x, we have

$$S(x') \simeq S(x) + \nabla S(x)^\top (x' - x), \tag{2}$$

for x' sufficiently close to x. As a result, if $\|\nabla S(x)\|$ is small, then equation (2) hints that the difference $|S(x') - S(x)|$ would likely remain small, implying stability of the score manifold around the point x. To this end, we propose the following regularizer. The specific definition of $\mathcal{L}_{\nabla S}$ is a design choice, however, for the case of the energy loss, which is the main focus of this paper, we use the

following formulation.

$$\begin{aligned}\mathcal{L}_{\nabla S_{\text{En}}} &= \mathbb{E}_{(x_{\text{in}},y_{\text{in}})\sim\mathcal{D}_{\text{in}}}\|\mathbb{I}_{S_{\text{En}}(x_{\text{in}})\leq m_{\text{in}}}\nabla_{x_{\text{in}}}S_{\text{En}}(x_{\text{in}})\| \\ &+ \mathbb{E}_{(x_{\text{aux}},y_{\text{aux}})\sim\mathcal{D}_{\text{aux}}}\|\mathbb{I}_{S_{\text{En}}(x_{\text{aux}})\geq m_{\text{aux}}}\nabla_{x_{\text{aux}}}S_{\text{En}}(x_{\text{aux}})\|\end{aligned} \tag{3}$$

To elaborate on the intuition behind this choice, consider the following cases:

1. If an ID (OOD) sample is correctly detected, we would want the local area around it to be detected the same.
2. If an ID (OOD) sample is mis-detected as OOD (ID), we do not want its local area to behave the same.

Hence, we select thresholds such that the loss only penalizes the gradient of the correctly detected samples. This is in sharp contrast to (1), where the aim is to penalize the energy score of the samples that have undesired energy levels. Therefore, by designing the gradient loss as (3), we allow the energy loss to train the samples that are not yet well learned and only use the samples whose score values are properly learned. Having specified $\mathcal{L}_{\nabla S}$, we train our model using the following loss $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_S\mathcal{L}_S + \lambda_{\nabla S}\mathcal{L}_{\nabla S}$, where λ_S and $\lambda_{\nabla S}$ are regularization constants and $\mathcal{L}_{\text{CE}}$ is the well-known cross-entropy loss.

4.2 OOD Sampling

One concern faced by OOD training schemes is that the OOD data may outnumber the ID data by orders of magnitude. Having access to such a large auxiliary dataset begs the question of which samples should the network see during the training phase. This creates two possible scenarios. Either the training algorithm exposes the model to an equal number of both ID and OOD data points, which means that the model cannot fully utilize the OOD dataset as most of the OOD dataset remains unseen. Alternatively, the algorithm may provide disproportionately more OOD samples, leading to bias in the model due to the imbalance in the samples [49].

These observations hint towards employing a more informative sampling strategy [6, 27] that utilizes as much of the dataset as possible while avoiding the pitfalls caused by imbalanced datasets. Whilst the first option that comes to mind is to perform a greedy sampling based on the score function S, it has been observed [22] that such greedy samples could bias the model to some specific regions of the image space, resulting in a model unable to generalize well.

To address these issues, we present Algorithm 1, which uses clustering (to discourage greedy behavior) alongside energy-based scoring to sample more informative images from the dataset. Next, we explain the sampling algorithm in more detail.

Clustering: Given samples $\{x_i\}_{i=1}^{n_{\text{OOD}}}$, we first calculate the features $z_i = h(x_i)$ and the scores $s_i = -\text{LSE}(f(x_i))$. The next step is to perform clustering in the *feature space*. To perform clustering, we use K-Means [32] with a fixed number

Algorithm 1 Energy-Based Sampling

Input: Auxiliary dataset $\mathcal{X} \sim \mathcal{D}_{\text{aux}}$ with n_{OOD} samples, number of clusters k , feature extractor h, score function S
Output: Selected samples: $\bar{\mathcal{X}} \subset \mathcal{X}$;
$z_i \leftarrow h(x_i)$, $s_i \leftarrow S(x_i), i = 1, \cdots, n_{\text{OOD}}$.
$(\hat{z}_1, \cdots, \hat{z}_{n_{\text{OOD}}}) \leftarrow h_N(z_1, \cdots, z_{n_{\text{OOD}}})$
Labels $l_i \leftarrow$ K-Means$(\{\hat{z}_i\}_{i=1}^{n_{\text{OOD}}}, k)$.
for $j = 1 \cdots k$ **do**
    $\mathcal{I} \leftarrow \{i : l_i = j\}$
    $\bar{\mathcal{X}}_j \leftarrow$ Sample $\{x_i\}_{i \in I}$ based on $\{s_i\}_{i \in I}$
end for
return $\bar{\mathcal{X}} \leftarrow \cup_{j=1}^{k} \bar{\mathcal{X}}_j$

of clusters. As K-Means assigns clusters based on Euclidean distances, we normalize the features to $\hat{z}_i = \frac{z_i}{\|z_i\|_2}$ to avoid skewed clusters towards samples with disproportionate feature values.

As for the number of clusters, we use the number of samples in each mini-batch of training, i.e., if in a training iteration the model is presented with n_{ID} ID samples, we cluster the n_{OOD} OOD samples into $k = n_{\text{ID}}$ clusters. We study the choice of the number of clusters in the Supplementary Material. Finally, given a clustering $\{C_j\}_{j=1}^{k}$, we can acquire cluster labels l_i for each data point $x_i, \quad i = 1, \cdots, n_{\text{OOD}}$.

Energy-Based Sampling: The final step in our sampling algorithm is to choose the *best* samples from each cluster with respect to some criterion. As we use the energy scores s_i to perform OOD detection, we use the same scores for sample selection. Given our choice of loss functions in (1) and (3), we use the samples with the smallest energy scores s_i of each cluster to provide samples useful for the loss term (1), and we use the samples with the largest energy scores to provide samples that are useful for the loss term (3). Intuitively, $\mathcal{L}_{\mathcal{S}_{\text{En}}}$ aims to increase the energy score of the OOD samples, and so, we need to provide samples whose energy scores are small. On the other hand, $\mathcal{L}_{\nabla \mathcal{S}_{\text{En}}}$ focuses on samples that already have high energy scores to make the model locally behave similarly, and so, we need to also provide samples with high energy scores.

Overall, by clustering, we ensure that the model is exposed to *diverse* regions of the feature space, and by sampling using the scores we ensure that from each region more *informative* samples are used.

5 Theoretical Analysis

In this section, we focus on the theoretical implications of gradient regularization. Traditionally, during the training phase, there is no control over the smoothness of the network and the score function, which can lead to overly sensitive OOD detection. The smoothness of networks has been studied in depth in certified robustness [12, 21, 33, 45] and has been deemed as a desired or even necessary property for the robustness of neural networks. The desirable properties sought by smoothness consequently follow over to OOD detection; simply because OOD

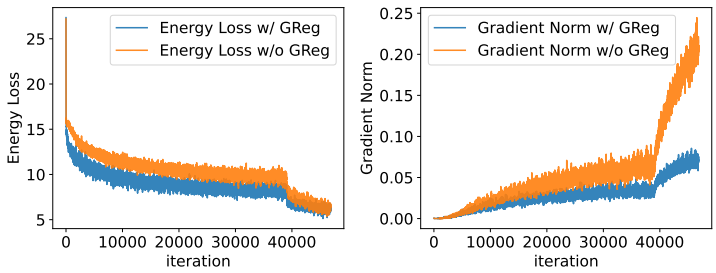


Fig. 3: The evolution of the Energy loss and the norm of the gradient with respect to the input of the score function during training with and without gradient regularization. Gradient regularization reduces the slope of the increase of the gradient norm, without negatively affecting the reduction in energy loss. The sudden change in the behavior of the plots towards the final iterations is due to learning rate scheduling.

detection can be cast as a simple classification task that the aforementioned literature studies. Intuitively, the lack of smoothness decreases the network's OOD robustness as the score values can abruptly change with small perturbations in the image space. To support our intuition, consider Figure 3 which portrays the evolution of the energy loss and the norm of its gradient with respect to the input in two cases, with and without *GReg*. Although the energy loss decreases in a similar fashion in both scenarios, the norm of the gradient increases much more rapidly in the absence of gradient regularization. The excess increase in the average norm of the gradient potentially points to a more sensitive score manifold. Note that the general increasing trend of the norm of the gradient in Figure 3 is not unexpected as the network is initialized randomly.

We further motivate our idea using the concept of penalizing the norm of the gradient [4] or upper bounds on the norm of the gradient, like the Lipschitz constant [12, 21] for achieving robustness. In other words, if the gradient of the score function S is bounded around some arbitrary point x, there exists an ε-neighborhood in which our prediction of ID/OOD would not change. This means that if the point x is detected as ID (OOD), then all the points $x' \in \mathcal{B}(x, \varepsilon)$ are also provably ID (OOD). This is desirable for us as it is usually assumed that the ID and OOD data are well separated. Formally, suppose S satisfies the following local Lipschitz property on $\mathcal{B}(x, \varepsilon)$ where $x \sim \mathcal{D}_{\text{in}}$

$$|S(x') - S(x)| \leq L_S \|x' - x\|, \qquad \forall x' \in \mathcal{B}(x, \varepsilon),$$

where $L_S > 0$. Suppose $x \sim \mathcal{D}_{\text{in}}$ is correctly classified by the score function as ID, i.e., $S(x) < \gamma$. If we want x' to be also labeled as ID, it is sufficient to satisfy

$$S(x) + L_S \|x' - x\| \leq \gamma,$$

which in turn gives the following certificate

$$\|x' - x\| \leq \varepsilon^* = \min\{\varepsilon, \frac{\gamma - S(x)}{L_S}\}, \tag{4}$$

i.e., all points x' within ε^* distance to x will also be labeled correctly as ID. A similar argument follows for OOD samples. This bound suggests that we can increase the certified radius by controlling L_S during training, e.g., through penalizing L_S. There exists a vast literature on Lipschitz estimations for neural networks [12,13,23]. However, using these methods on deep models increases the computational load of training. Instead, we penalize the norm of the gradient, which can be considered as a proxy to the local Lipschitz constant.

Furthermore, for piecewise-linear networks like ReLU and LeakyReLU networks, the local Lipschitz constant is equal to the norm of the gradient. More specifically, for every point x*, there exists an $\hat{\varepsilon} > 0$ such that the network remains linear in this region, i.e.,

$$f(x') = Ax' + b, \qquad \forall x' \in \mathcal{B}(x, \hat{\varepsilon}).$$

In such a region, the local Lipschitz constant is equal to the norm of the gradient of the points [3], i.e., $L_f = \|A\|$. This further motivates directly penalizing the norm of the gradient of S. Thus, for such networks, minimizing $\|\nabla S(x)\|$ decreases the local Lipschitz constant in the neighborhood of x, which in turn could increase the maximum certified perturbation radius in (4). Note that our argument for using the norm of the gradient in practice does not require the knowledge of $\hat{\varepsilon}$. We just use the fact that such an $\hat{\varepsilon}$ exists and that in the corresponding region, the network will be linear, and as a result, minimizing the norm of the gradient is equivalent to minimizing the local Lipschitz constant for this local neighborhood.

Post-hoc Methods and Gradient Regularization: So far, we have discussed the idea of gradient regularization and provided a certified radius for which the model is robust against OOD misdetection using the Lipschitz constant of the network. This radius (4) is proportional to the inverse of the Lipschitz constant of the score function, and by regularizing the gradient we aim to reduce the Lipschitz constant to increase this radius. Here, we note that many of the state-of-the-art OOD detection methods can also be analyzed in this framework. For example, DICE [47] and LINe [1] use activation pruning and sparsification to remove noisy neurons. It has been shown [21] that for a given matrix, removing a row or column of it results in a reduction of the Lipschitz constant. As activation pruning is essentially a form of masking the weights of the network, methods utilizing activation pruning reduce the upper bound on the Lipschitz constant. By doing so, they acquire a network with a smaller Lipschitz upper bound that provides a larger certified radius.

6 Experiments

We perform extensive OOD detection experiments on CIFAR [25] and ImageNet [9] benchmarks. In our comparisons, we use a wide range of CNN-based

*We discard the negligible case of points lying on the boundary of several piecewise-linear spaces.

Table 1: Comparison on the CIFAR-10 benchmark. Mean AUROC and FPR95 percentages on various benchmarks. The experimental results are reported over three trials. The best mean results are bolded and the runner-up is underlined. AUROC and FPR95 are percentages.

Dataset	Method	ResNet		WRN		DenseNet	
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
CIFAR-10	MSP [ICLR17] [17]	58.04	90.49	51.52	90.8	52.56	91.7
	ODIN [ICLR18] [28]	34.46	91.59	35.23	89.82	24.88	94.42
	Energy Score [Neurips20] [30]	39.32	92.24	33.4	91.76	28.99	94.09
	ReAct [NeurIPS21] [46]	39.44	92.30	37.54	91.49	25.83	95.27
	Dice [ECCV22] [47]	42.40	91.25	34.12	91.74	26.37	94.61
	LINe [CVPR23] [1]	45.25	90.59	36.27	90.2	14.84	96.95
	OE [ICLR18] [18]	19.92	95.71	21.12	95.55	21.76	95.8
	Energy Loss [NeurIPS20] [30]	<u>11.14</u>	<u>97.53</u>	<u>13.11</u>	<u>97.14</u>	<u>11.26</u>	<u>97.43</u>
	OpenMix [CVPR23] [55]	22.24	96.26	21.92	96	22.86	95.65
	GReg	**7.9**	**97.95**	**7.95**	**98.1**	**7.93**	**98.12**

Table 2: Comparison on the CIFAR-100 benchmark. Mean AUROC and FPR95 percentages on various benchmarks. The experimental results are reported over three trials. The best mean results are bolded and the runner-up is underlined.

Dataset	Method	ResNet		WRN		DenseNet	
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
CIFAR-100	MSP [ICLR17] [17]	74.66	80.10	80.39	75.37	85.87	68.78
	ODIN [ICLR18] [28]	64.00	84.33	67.95	79.86	69.97	81.55
	Energy Score [Neurips20] [30]	68.46	84.15	74.15	79.26	78.91	76.12
	ReAct [NeurIPS21] [46]	59.38	87.59	70.65	80.2	76.27	80.1
	Dice [ECCV22] [47]	77.29	80.35	72.32	79.62	69.80	80.50
	LINe [CVPR23] [1]	72.09	82.93	67.5	80.9	35.16	88.72
	OE [ICLR18] [18]	77.32	63.89	74.22	65.70	60.52	84.21
	Energy Loss [NeurIPS20] [30]	62.77	84.05	65.62	80.18	64.37	83.86
	POEM [ICML22] [35]	61.11	82.43	62.58	79.74	59.33	79.81
	OpenMix [CVPR23] [55]	59.64	88.09	73.12	79.23	67.29	84.47
	DOS [ICLR24] [22]	<u>54.62</u>	<u>88.30</u>	**45.26**	<u>90.76</u>	<u>34.92</u>	<u>93.22</u>
	GReg	59.6	82.92	58.26	86.56	56.29	87.35
	GReg+	**50.78**	**88.75**	<u>48.12</u>	**91.02**	**30.55**	**93.38**

architectures. We also provide ablation analysis to demonstrate the effect of each component within our framework.

6.1 Setup

Datasets. For CIFAR experiments, we use CIFAR-10 and CIFAR-100 as ID datasets. We evaluate the models on six different OOD datasets including Textures [8], SVHN [38], Places365 [54], LSUN-cropped, and LSUN-resized [52]. Following [47], at test time, we use 10000 randomly selected samples from each of these 6 datasets. Following [55], instead of the recently retracted TinyImages, we use the 300K RandomImages [18] as the auxiliary outlier dataset.

For the ImageNet experiments, following [22], we consider a set of 10 *random* classes from ImageNet-1K as the ID dataset and the rest of the dataset as the auxiliary outlier dataset. For evaluation, we use 10000 randomly selected samples from Textures, Places, SUN [51], and iNaturalist [48] as OOD datasets.

Experimental Setup: For the experiments of *GReg* on CIFAR, we use a pre-trained model of the corresponding architecture and finetune for 20 epochs using

Table 3: Comparison on the ImageNet benchmark. The experimental results are reported over three trials. The best mean results are bolded and the runner-up is underlined. AUROC and FPR95 are percentages.

Method	OOD Datasets								Average	
	iNaturalist		SUN		Places		Textures			
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
MSP [ICLR17] [17]	62.6	87.09	64.3	86.64	62.3	86.72	78.3	75.03	66.87	83.87
ODIN [ICLR18] [28]	48.5	92.07	53.9	90.77	49.3	91.06	69.6	80.77	55.32	88.66
Energy Score [Neurips20] [30]	54	90.77	52.5	90.7	46.6	91.37	73.1	77.54	56.55	87.59
ReAct [NeurIPS21] [46]	51.88	91.52	52.48	90.81	49.56	90.92	64.61	83.77	54.63	89.25
DICE [ECCV22] [47]	32.86	93.6	41.35	92.9	45.82	91.58	65.2	78.61	46.3	89.17
LINe [CVPR23] [1]	29.47	93.77	37.29	92.85	38.85	92.16	52.32	86.38	<u>39.48</u>	91.29
OE [ICLR18] [18]	34.56	92.75	54.63	89.75	54.86	89.05	76.2	75.55	55.06	86.78
Energy Loss [Neurips20] [30]	18.44	95.91	50.19	90.37	49.32	90.47	62.96	77.78	45.23	88.63
DOS [ICLR24] [22]	61.63	88.68	42.69	93.23	45.49	93.37	47.45	92.63	49.31	<u>91.97</u>
GReg	29.54	94.94	45.86	92.53	45.35	92.06	68.29	80.98	47.26	90.13
GReg+	24.11	95	33.98	92.31	32.89	92.7	49.32	88.22	**35.08**	**92.06**

SGD with cosine annealing. For the experiments of *GReg+* on CIFAR, we train a model from scratch for 50 epochs with a learning rate of 0.1 and then further train for another 10 epochs with a learning rate of 0.01, using the same optimizer. Following [30], we set $\lambda_S = 0.1$ and set $\lambda_{\nabla S} = 1$.
For the ImageNet experiment we only perform finetuning on a pre-trained DenseNet-121 [19] due to limited resources. See the Supplementary Material for more details.
For the CIFAR experiments, we evaluate our results on ResNet-18 [15], WRN-40 [53], and DenseNet-101 [19] to span a wide range of architectures and parameter numbers.

Sampling Details: Following the explanations in Section 4.2, we cluster the auxiliary samples into n_{ID} clusters and choose the lowest and highest scoring samples of each cluster, i.e., in each iteration, we present $2n_{\text{ID}}$ auxiliary samples for the training of that iteration. For the CIFAR experiments, we use the whole auxiliary dataset in each epoch for clustering. In other words, at each epoch, the 300K images are split into mini-batches, and clustering is performed on each mini-batch at each iteration so that no auxiliary sample is missed. This means the clustering is performed on each mini-batch and not the full auxiliary dataset, as doing so would be computationally prohibitive. For the ImageNet experiment, in each epoch, we sample 600K randomly selected images from the remaining (990) classes as the auxiliary dataset and perform the clustering on each mini-batch.

Evaluation Metrics: We report the following metrics: 1) *FPR95*: the false positive rate of the OOD samples when the true positive rate of ID samples is at 95%. 2) *AUROC*: the area under the receiver operating characteristic curve.

Comparison Methods: For comparison with post-hoc OOD methods, we compare to MSP [17], ODIN [18], Energy (score) [30], ReAct [46], DICE [47] and LINe [1]. For the methods that use auxiliary data, we compare with OE [18], Energy (loss) [30], POEM [35], OpenMix [55] and DOS [22]. MSP uses the maximum softmax probability to detect the OOD samples. ODIN improves upon MSP by

Table 4: Ablation Study of *GReg*. AUROC and FPR95 percentages on CIFAR benchmarks averaged over all OOD datasets and three runs on DenseNet.

Method	CIFAR-10		CIFAR-100	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
OE	21.76	95.8	**60.52**	84.21
OE + Grad	**18.79**	**96.32**	62.54	**84.71**
Energy	11.26	97.43	64.37	83.86
Energy + Grad	**7.93**	**98.12**	**56.29**	**87.35**
POEM	29.15	94.18	59.33	79.81
POEM + Grad	**23.87**	**95.41**	**47.94**	**86.63**

introducing noise and temperature scaling. Energy proposes a score function for OOD detection. ReAct uses activation clipping. DICE and LINe build up on that idea by also detecting the important neurons and activation to improve OOD detection. OE and Energy train based on their respective loss functions. POEM uses posterior sampling and DOS proposes diverse sampling to improve OOD detection. OpenMix uses pseudo-samples to improve misclassification detection.

6.2 Results

Evaluation on CIFAR-10: We first evaluate the performance of our method on the CIFAR-10 dataset. Following the literature [1], Table 1 shows the average results over the six commonly used OOD datasets. The detailed results on individual OOD datasets can be found in the Supplementary Material. As the results on CIFAR-10 are saturated, in Table 1 we focus on the effectiveness of gradient regularization and report the rest of the methods in the Supplementary Material. It can be seen that *GReg* outperforms all the methods in both of the reported metrics. Furthermore, on CIFAR-10, the use of local information through gradient regularization, allows *GReg* to decrease the average FPR95 of Energy [30] by 3.9%.

Evaluation on CIFAR-100: For the results of this part, we use the same experimental setup and only change the ID dataset to CIFAR-100. The average results of this experiment set are presented in Table 2 and the individual results can be found in the Supplementary Material. In this setup, *GReg* provides competitive results on the ResNet and WRN architectures with state-of-the-art methods that do not employ sampling. Moreover, coupled with sampling, *GReg+* outperforms all current state-of-the-art methods on almost all metrics of the different architectures. On average, *GReg+* outperforms the FPR95 metric of DOS [22] by 1.8%, which showcases the effectiveness of energy-based sampling. The detailed results on each OOD dataset can be found in the Supplementary Material.

Evaluation on ImageNet: For this experiment, we randomly select 10 classes of Imagenet-1K as ID and use the rest as the auxiliary data. The results of this large-scale experiment are presented in Table 3. We can see that in terms of FPR95, *GReg+* outperforms DOS and LINe by large margins of 14.23% and 3.68%, respectively, while achieving state-of-the-art AUROC performance. The

energy-based sampling approach enables our method to acquire more informative samples compared to the sampling method in [22], resulting in improved FPR95.

6.3 Ablation Study

In this section, we perform extensive experiments to show the effectiveness of components within our framework, individually. That is, we show the effect of regularizing the gradient on other methods and then compare our proposed sampling method with the state-of-the-art. We also study the effect of our OOD schemes on the ID accuracy of the models.

Gradient Regularization: Due to the generality of the idea of gradient regularization, our method can be used in conjunction with many of the existing methods that require training. These methods include but are not limited to, Energy [30], OE [18], POEM [35], etc. In order to leverage the gradient regularization idea, one needs to define a suitable loss function based on the norm of the gradient of the score being used by the method. Once the gradient loss is designed, adding this term to the total loss function promotes a smoother score manifold learned by the neural network.

To show the effectiveness of gradient regularization on different OOD training schemes, we choose three of the most well-known OOD detection methods, Energy, OE, and POEM, and train a DenseNet with and without the gradient regularization loss on both of the CIFAR benchmarks. Table 4 shows the overall improvement of all these methods with gradient regularization. This improvement, in both the FPR95 and AUROC, supports our claim that regularizing the gradient improves the OOD detection performance and that coupling *GReg* with other methods could improve their performance.

Sampling Methods: To study the effect of various sampling methods, we compare POEM and DOS to our method with and without gradient regularization. We denote *GReg+* without gradient regularization as Energy-only-clustering. Table 5 compares different methods that utilize some form of sampling and shows the impact of energy-based clustering. On top of that, the comparison between Energy-only-clustering and *GReg+* further encourages the use of gradient regularization.

ID Accuracy: To study the effect of gradient regularization on accuracy, we compare the ID accuracy of our methods to the most relevant state-of-the-art on WRN and DenseNet architectures to ensure that our method does not have a large negative impact on the ID accuracy. Table 6 shows that gradient regularization (*GReg*) improves the ID accuracy of energy on both architectures and *GReg+* also has superior ID accuracy in comparison with DOS.

7 Conclusions

In this paper, we presented the idea of gradient regularization of the score function in OOD detection. We also developed an energy-based sampling algorithm

Table 5: Ablation Study on Sampling. Comparison among sampling methods on CIFAR-100 on DenseNet averaged over six OOD datasets.

Method	CIFAR-100	
	FPR95 ↓	AUROC ↑
POEM	59.33	79.81
DOS	34.92	93.22
Energy only clustering	31.77	93.20
GReg+	**30.55**	**93.38**

Table 6: Ablation Study on ID Accuracy. Accuracy comparison on CIFAR-10 on two architectures.

Method	CIFAR-10	
	WRN	DenseNet
Clean	94.84	94.03
Energy loss	88.15	91.82
GReg	89.55	92.14
DOS	78.90	87.06
GReg+	83.06	89.89

to improve sampling quality when the auxiliary dataset is large. We demonstrate that *GReg* exhibits superior performance compared to methods that do not require clustering, and *GReg+* outperforms all state-of-the-art methods. Furthermore, our experiments show that regularizing the gradient could result in robust models capable of extracting local information from the ID and OOD data. We also provide detailed analytical analysis and ablation studies to support our method.

References

1. Ahn, Y.H., Park, G.M., Kim, S.T.: Line: Out-of-distribution detection by leveraging important neurons. arXiv preprint arXiv:2303.13995 (2023) 2, 4, 10, 11, 12, 13, 21, 22
2. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1563–1572 (2016) 3
3. Bhowmick, A., D'Souza, M., Raghavan, G.S.: Lipbab: Computing exact lipschitz constant of relu networks. In: Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part IV 30. pp. 151–162. Springer (2021) 10
4. Chan, A., Tay, Y., Ong, Y.S., Fu, J.: Jacobian adversarially regularized networks for robustness. arXiv preprint arXiv:1912.10185 (2019) 9
5. Chen, G., Qiao, L., Shi, Y., Peng, P., Li, J., Huang, T., Pu, S., Tian, Y.: Learning open set network with discriminative reciprocal points. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 507–522. Springer (2020) 3
6. Chen, J., Li, Y., Wu, X., Liang, Y., Jha, S.: Atom: Robustifying out-of-distribution detection using outlier mining. In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21. pp. 430–445. Springer (2021) 2, 3, 4, 7
7. Choi, H., Jeong, H., Choi, J.Y.: Balanced energy regularization loss for out-of-distribution detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15691–15700 (2023) 4, 6

8. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3606–3613 (2014) 11
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 10
10. Dhamija, A.R., Günther, M., Boult, T.: Reducing network agnostophobia. Advances in Neural Information Processing Systems **31** (2018) 4
11. Du, X., Wang, Z., Cai, M., Li, Y.: Vos: Learning what you don't know by virtual outlier synthesis. arXiv preprint arXiv:2202.01197 (2022) 4
12. Fazlyab, M., Entesari, T., Roy, A., Chellappa, R.: Certified robustness via dynamic margin maximization and improved lipschitz regularization (2023) 8, 9, 10
13. Fazlyab, M., Robey, A., Hassani, H., Morari, M., Pappas, G.: Efficient and accurate estimation of lipschitz constants for deep neural networks. In: Advances in Neural Information Processing Systems. pp. 11427–11438 (2019) 10
14. Ge, Z., Demyanov, S., Chen, Z., Garnavi, R.: Generative openmax for multi-class open set classification. arXiv preprint arXiv:1707.07418 (2017) 4
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 12
16. Hein, M., Andriushchenko, M., Bitterwolf, J.: Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 41–50 (2019) 3, 5
17. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=Hkg4TI9xl 2, 3, 4, 11, 12, 21, 22
18. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HyxCxhRcY7 2, 4, 6, 11, 12, 14, 21, 22
19. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017) 12
20. Huang, R., Geng, A., Li, Y.: On the importance of gradients for detecting distributional shifts in the wild. Advances in Neural Information Processing Systems **34**, 677–689 (2021) 4
21. Huang, Y., Zhang, H., Shi, Y., Kolter, J.Z., Anandkumar, A.: Training certifiably robust neural networks with efficient local lipschitz bounds. Advances in Neural Information Processing Systems **34**, 22745–22757 (2021) 8, 9, 10
22. Jiang, W., Cheng, H., Chen, M., Wang, C., Wei, H.: Dos: Diverse outlier sampling for out-of-distribution detection. arXiv preprint arXiv:2306.02031 (2023) 2, 3, 4, 6, 7, 11, 12, 13, 14, 22
23. Jordan, M., Dimakis, A.G.: Exactly computing the local lipschitz constant of relu networks. Advances in Neural Information Processing Systems **33**, 7344–7353 (2020) 10
24. Kong, S., Ramanan, D.: Opengan: Open-set recognition via open data generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 813–822 (2021) 4
25. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 10

26. Li, J., Chen, P., He, Z., Yu, S., Liu, S., Jia, J.: Rethinking out-of-distribution (ood) detection: Masked image modeling is all you need. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11578–11589 (2023) 3
27. Li, Y., Vasconcelos, N.: Background data resampling for outlier-aware classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13218–13227 (2020) 7
28. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:1706.02690 (2017) 2, 11, 12, 21, 22
29. Lin, Z., Roy, S.D., Li, Y.: Mood: Multi-level out-of-distribution detection. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 15313–15323 (2021) 4
30. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Advances in neural information processing systems **33**, 21464–21475 (2020) 2, 4, 5, 6, 11, 12, 13, 14, 19, 21, 22
31. Liu, X., Lochman, Y., Zach, C.: Gen: Pushing the limits of softmax-based out-of-distribution detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23946–23955 (2023) 2, 3, 4
32. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982) 4, 7
33. Losch, M., Stutz, D., Schiele, B., Fritz, M.: Certified robust models with slack control and large lipschitz constants. arXiv preprint arXiv:2309.06166 (2023) 8
34. Ming, Y., Cai, Z., Gu, J., Sun, Y., Li, W., Li, Y.: Delving into out-of-distribution detection with vision-language representations. Advances in Neural Information Processing Systems **35**, 35087–35102 (2022) 3
35. Ming, Y., Fan, Y., Li, Y.: Poem: Out-of-distribution detection with posterior sampling. In: International Conference on Machine Learning. pp. 15650–15665. PMLR (2022) 2, 3, 4, 6, 11, 12, 14, 22
36. Moon, W., Park, J., Seong, H.S., Cho, C.H., Heo, J.P.: Difficulty-aware simulator for open set recognition. In: European Conference on Computer Vision. pp. 365–381. Springer (2022) 4
37. Neal, L., Olson, M., Fern, X., Wong, W.K., Li, F.: Open set learning with counterfactual images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 613–628 (2018) 4
38. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011) 11
39. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 427–436 (2015) 1
40. Olber, B., Radlak, K., Popowicz, A., Szczepankiewicz, M., Chachuła, K.: Detection of out-of-distribution samples using binary neuron activation patterns. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3378–3387 (2023) 3
41. Oza, P., Patel, V.M.: C2ae: Class conditioned auto-encoder for open-set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2307–2316 (2019) 3
42. Park, J., Park, H., Jeong, E., Teoh, A.B.J.: Understanding open-set recognition by jacobian norm of representation. arXiv preprint arXiv:2209.11436 (2022) 4

43. Safaei, B., Vibashan, V., de Melo, C.M., Hu, S., Patel, V.M.: Open-set automatic target recognition. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023) 3
44. Shapley, L.S., et al.: A value for n-person games (1953) 4
45. Srinivas, S., Matoba, K., Lakkaraju, H., Fleuret, F.: Efficient training of low-curvature neural networks. Advances in Neural Information Processing Systems **35**, 25951–25964 (2022) 8
46. Sun, Y., Guo, C., Li, Y.: React: Out-of-distribution detection with rectified activations. Advances in Neural Information Processing Systems **34**, 144–157 (2021) 4, 11, 12, 21, 22
47. Sun, Y., Li, Y.: Dice: Leveraging sparsification for out-of-distribution detection. In: European Conference on Computer Vision. pp. 691–708. Springer (2022) 2, 4, 10, 11, 12, 21, 22
48. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8769–8778 (2018) 11
49. Wang, L., Han, M., Li, X., Zhang, N., Cheng, H.: Review of classification methods on unbalanced data sets. IEEE Access **9**, 64606–64628 (2021) 7
50. Wang, Q., Ye, J., Liu, F., Dai, Q., Kalander, M., Liu, T., Hao, J., Han, B.: Out-of-distribution detection with implicit outlier transformation. arXiv preprint arXiv:2303.05033 (2023) 3
51. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 3485–3492. IEEE (2010) 11
52. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015) 11
53. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016) 12
54. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017) 11
55. Zhu, F., Cheng, Z., Zhang, X.Y., Liu, C.L.: Openmix: Exploring outlier samples for misclassification detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12074–12083 (June 2023) 4, 6, 11, 12, 21, 22

8 Supplementary Materials

In the supplementary materials, we provide detailed implementation details and experimental results for both CIFAR and ImageNet benchmarks. Furthermore, we present an analysis to show the effectiveness of our OOD sampling strategy. We then analyze the energy distribution of the ID and OOD samples.

8.1 Additional Implementation Details

For the experiments on CIFAR we use SGD with a momentum of 0.9, and a weight decay of 0.0001. For *GReg* on CIFAR, similar to [30], we decrease the learning rate following a cosine annealing strategy, with a maximum learning rate of 1 and a minimum of 0.001. We use a batch size of 64 and 32 for the CIFAR and ImageNet experiments, respectively.
For both *GReg* and *GReg+* on the ImageNet dataset we perform fine-tuning of a pre-trained DenseNet-121 model (in contrast to the CIFAR benchmarks where we train *GReg+* from scratch) using the ADAM optimizer with an initial learning rate of 10^{-4} and decrease the learning rate to 10^{-5} at epoch 10. We then run *GReg* and *GReg+* to reach epochs 20 and 15, respectively. The other hyperparameters are the same as the CIFAR benchmarks.

To reproduce the results of MSP, ODIN, and Energy, we used the codebase of Energy*. For the case of ReAct and DICE, we utilize the codebase of DICE*, and to reproduce the results of LINe*, and DOS*, we utilize their corresponding codebases. In the spirit of fairness, we run their codes with multiple specifications similar to the original manuscript and report the best results in our tables. Furthermore, experiments requiring training are conducted with 3 different seeds and we report the average values in the tables. All other methods are run with their corresponding default parameters outlined in their manuscript.

8.2 Detailed CIFAR-10/100 Benchmark Results

Table 8 and Table 9 show the complete and detailed performance of various OOD detection approaches on each of the six OOD test datasets. Table 10 compares the sampling methods on the CIFAR-10 dataset. As it can be seen, especially on the WRN and DenseNet architectures, the performance of different methods is saturated.

8.3 Distribution Analysis

To further study the effectiveness of our method, we perform the following analysis. We choose a WRN model pre-trained on CIFAR-10 and fine-tune it with

*https://github.com/wetliu/energy_ood
*https://github.com/deeplearning-wisc/dice
*https://github.com/YongHyun-Ahn/LINe-Out-of-Distribution-Detection-by-Leveraging-Important-Neurons
*https://github.com/lygjwy/DOS

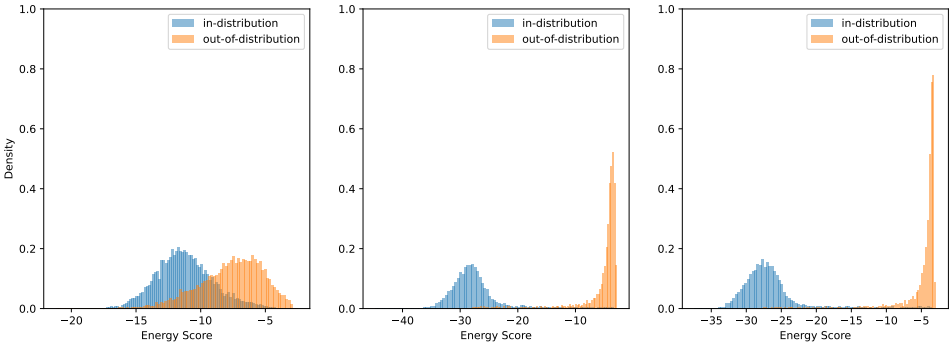


Fig. 4: Distributions of energy scores. The left figure shows the distribution of the pre-trained model, the middle shows the same for Energy loss and the right figure shows the distribution for *GReg*.

Table 7: Ablation study on the number of clusters. FPR95 and AUROC percentages reported on CIFAR benchmarks on DenseNet.

Num. of Clusters	CIFAR-10		CIFAR-100	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
No Clustering	**7.93**	**98.12**	56.29	87.35
16	9.88	97.68	58.97	82.88
32	12.1	97.15	46.41	88.433
64	11.27	97.54	**30.55**	**93.38**

the energy loss and *GReg*. Next, we plot the distribution of energy scores for ID (CIFAR-10 test) and OOD (SVHN) datasets and compare the results. As it can be seen from Figure 4, adding gradient regularization to the energy loss enables the network to better distinguish ID samples from the OOD since the distance between the two distributions has increased, i.e., the OOD data are assigned higher energy scores, resulting in further distanced score distributions.

8.4 Sampling

In our setting, we are presented with three main possible clustering spaces: the image space, the feature space, or the logit space. The image space contains the most information but there are two main drawbacks to using it. Firstly, the image space is far too large and high-dimensional, typically in the order of tens of thousands. Secondly, although all the information is in the image, no processing has been done on the image and the features have not been extracted, meaning that in effect, there is also a lot of noise irrelevant to our task. The logit space is not suitable from the opposite perspective, i.e., most of the information has been stripped and only label-relevant information remains. The feature space presents a sweet spot; reasonable dimensionality, relevant features, and smaller levels of noise. Consequently, we choose to cluster the samples in the *feature space*.

Another parameter of interest in the experiments is the number of clusters. In Table 7 we examine the effect of the number of clusters on the performance

Table 8: Comparison on the CIFAR-10 benchmark. The experimental results are reported over three trials. AUROC and FPR95 are percentages.

Network	Method	OOD Datasets												Average	
		Textures		SVHN		Places		LSUN-c		LSUN-r		iSUN			
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
ResNet	MSP [17]	58.78	90.03	73.02	89.42	61.56	87.98	43.88	94.21	53.31	91.38	57.73	89.95	58.04	90.49
	ODIN [28]	50.42	86.89	40.27	91.84	48.65	86.56	7.81	98.21	26.48	94	33.14	92.06	34.46	91.59
	Energy score [30]	53.26	89.12	58.36	91.01	44.51	89.45	12.57	97.64	30.48	94.11	36.77	92.16	39.32	92.24
	ReAct [46]	52.04	89.68	60.2	90.46	44.68	89.26	13.36	97.51	29.95	94.29	36.43	92.63	39.44	92.30
	DICE [47]	54.73	88.34	59.4	89.81	44.09	89.39	22.38	95.72	33.32	93.17	40.52	91.08	42.40	91.25
	LINe [1]	56.97	88.07	66.57	87.4	46.05	88.62	24.9	95.13	34.66	93.12	42.38	91.21	45.25	90.59
	OE [18]	19.46	95.98	13.94	97.22	38.38	90.76	8.15	98.20	21.03	95.87	18.58	96.26	19.92	95.71
	Energy Loss [30]	12.95	97.17	4.22	98.89	27.31	94.05	5.47	98.69	7.87	98.24	9.04	98.16	11.14	97.53
	OpenMix [55]	17.76	96.77	41.12	94.44	27.68	94.4	8.33	98.32	19.06	96.87	19.54	96.79	22.24	96.26
	GReg	5.96	98.36	3	98.83	22.76	94.8	3.03	99	6.5	98.4	6.2	98.33	7.91	97.95
WRN	MSP [17]	59.53	88.45	48.53	91.74	59.86	88.29	31.15	95.6	53.22	91.14	56.87	89.58	51.52	90.8
	ODIN [28]	54.52	80.45	46.6	85.68	48.57	86.04	10.19	97.84	22.86	95.05	28.64	93.87	35.23	89.82
	Energy score [30]	52.52	85.38	36.58	90.73	39.88	89.87	8.2	98.34	28.8	93.87	34.47	92.38	33.40	91.76
	ReAct [46]	53.47	86.58	41.46	89.34	41	90.51	13.21	97.4	34.95	93.30	41.15	91.82	37.54	91.49
	DICE [47]	58.97	84.24	46.09	88.22	42.55	89.69	2.87	99.27	23.6	95.29	30.66	93.78	34.12	91.74
	LINe [1]	57.13	83.81	50.2	83.11	47.35	87.67	2	99.47	27.32	94.32	33.65	92.82	36.27	90.2
	OE [18]	22.56	95.24	27.41	95.13	33.65	92.24	8.88	98.19	17.58	96.22	16.68	96.27	21.12	95.55
	Energy Loss [30]	11.47	97.3	23.93	94.93	22.6	95.37	5.07	98.83	8.07	98.17	7.53	98.27	13.11	97.14
	OpenMix [55]	21.17	95.85	26.63	95.65	27.17	94.13	14.32	97.35	21.8	96.42	20.45	96.61	21.92	96
	GReg	7.83	98.16	6.23	98.4	19.9	95.66	3.8	98.93	5.26	98.7	4.66	98.76	7.95	98.10
DenseNet	MSP [17]	66.3	87.09	44.64	93.86	63.16	88.35	43.34	94.17	48.9	93.4	49.05	93.35	52.56	91.70
	ODIN [28]	55.62	85.03	29.03	94.86	42.44	91.11	11.94	97.67	4.86	98.96	5.39	98.9	24.88	94.42
	Energy score [30]	60.03	85.17	35.2	94.76	40.9	91.58	9.5	98.17	13.78	97.51	14.57	97.4	28.99	94.09
	ReAct [46]	50.47	90.53	29.81	95.56	40.35	92.05	10.04	98.11	11.44	97.79	12.92	97.62	25.83	95.27
	DICE [47]	56.26	86.42	40.47	93.99	39.6	91.82	3.79	99.15	8.71	98.19	9.42	98.11	26.37	94.61
	LINe [1]	23.35	95.11	12.22	97.56	43.72	91.13	0.61	99.83	4.09	99.1	5.06	99.02	14.84	96.95
	OE [18]	20.93	96.04	12.19	97.68	39.18	91.69	7.15	98.57	25.45	95.42	25.72	95.41	21.77	95.8
	Energy Loss [30]	13.43	97.04	20.08	95.52	20.09	95.41	3.53	99.19	4.55	98.84	5.87	98.63	11.26	97.44
	OpenMix [55]	24.8	95.3	40.59	92.58	29.13	93.88	12.39	97.64	15.4	97.23	14.9	97.31	22.87	95.66
	GReg	8.3	97.9	3.7	98.83	20.13	95.66	3.1	99.23	6.36	98.53	6.03	98.56	7.93	98.12

of our method on both CIFAR benchmarks using the DenseNet architecture. It can be observed that our sampling significantly improves the results on the CIFAR-100 benchmark, but does not have the same effect on CIFAR-10. Our intuition suggests that one reason behind this observation might be that the CIFAR-10 benchmark is so simple that the DenseNet model can observe all the auxiliary data during the training phase and learn the information. However, in the case of CIFAR-100, as the benchmark is harder, sampling the data helps the model extract more informative samples. Note that the perplexity in comparing CIFAR-100 vs CIFAR-10 is that the datasets contain the same number of samples (60000 total images in both) but CIFAR-100 has 10 times the number of classes of CIFAR-10. This can also be seen in our ImageNet experiments (see Table 3). That is, as the task is harder, sampling using *GReg+* is more effective and improves upon *GReg* by large margins of 12% and 2% on FPR and AUROC, respectively.

Table 9: Comparison on the CIFAR-100 benchmark. The experimental results are reported over three trials. AUROC and FPR95 are percentages.

Network	Method	OOD Datasets												Average	
		Textures		SVHN		Places		LSUN-c		LSUN-r		iSUN			
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
ResNet	MSP [17]	82.18	75.44	70.19	82.53	82.09	75.06	64.37	85.43	74.74	81.24	74.39	80.94	74.66	80.1
	ODIN [28]	76.44	77.59	65.24	84.61	83.37	74.62	41.63	93.07	59.85	88	57.48	88.09	64	84.33
	Energy score [30]	81.82	77.71	57.08	89.09	84.12	75.4	50.24	91.71	69.73	85.49	67.77	85.5	68.46	84.15
	ReAct [46]	67.69	85.5	48.94	91.8	82.29	77.2	31.54	94.61	63.9	88.53	61.96	87.95	59.38	87.59
	DICE [47]	88.92	71.5	69.78	85.12	87.81	72.13	59.78	89.08	77.69	82.87	79.81	81.42	77.29	80.35
	LINe [1]	78.71	79.96	66.2	86.98	87.71	72.88	43.98	91.09	78.27	83.48	77.69	83.19	72.09	82.93
	OE [18]	73.91	73.14	86.45	70.18	69.28	77.75	44.44	86.51	94.55	38.80	95.30	36.98	77.32	63.89
	Energy Loss [30]	57.3	83.9	75.06	84.33	82.2	75.8	35.03	93.93	62.06	83.4	65	82.93	62.77	84.05
	POEM [35]	67.92	77.69	78.72	78.6	89.07	68.76	41.97	92.24	46.27	88.31	42.72	89.01	61.11	82.43
	OpenMix [55]	59.87	87.54	72.91	86.76	75.73	80	48.75	91.31	48.02	92.25	52.56	90.71	59.64	88.09
	DOS [22]	52.29	88.1	47.57	91.44	77.99	81.42	54.75	89	45.5	90.76	49.62	89.08	54.62	88.30
	GReg	58.73	83.73	42	91.7	78.4	76.7	28.56	94.6	74.56	75.2	75.33	75.63	59.6	82.92
	GReg+	48.74	88.2	42.2	92.45	69.61	83.35	43.86	90.75	46.04	89.94	54.22	87.8	50.78	88.75
WRN	MSP [17]	83.05	73.75	83.88	71.97	82.03	73.91	66.59	83.71	83.11	74.12	83.66	74.77	80.39	75.37
	ODIN [28]	76.57	75.13	90.37	67.08	81.12	74.21	36.44	93.32	62	84.55	61.23	84.89	67.95	79.86
	Energy score [30]	79.78	76.47	85.57	74.43	80.07	75.69	36.09	93.4	80.6	77.72	82.82	77.86	74.15	79.26
	ReAct [46]	67.12	82.97	74.45	88.28	81.21	74.24	36.63	91.67	81.23	72	83.31	72.07	70.65	80.20
	DICE [47]	85.32	72.84	87.21	71.39	81.02	74.95	16.73	96.83	79.86	81.64	83.82	80.08	72.32	79.62
	LINe [1]	67.34	81.98	81.73	83.6	84.03	71.03	16.93	96.61	77.98	76.26	77	76.37	67.50	80.97
	OE [18]	73.24	72.47	71.6	81.3	69.2	77.97	40.46	87.26	95.04	38.36	95.77	36.84	74.22	65.7
	Energy Loss [30]	68.96	80.66	87.7	73.6	83.26	74.36	55.53	88.53	47.7	81.96	50.6	82	65.62	80.18
	POEM [35]	79.23	70.17	90.37	67.89	85.89	70.51	26.09	94.35	45.3	88.59	48.59	86.92	62.58	79.74
	OpenMix [55]	64.91	83.61	87.25	70.34	73.17	78.14	59.18	87.77	75.19	78.46	79.02	77.05	73.12	79.23
	DOS [22]	41.89	91.58	15.07	97.33	59.18	88.24	32.12	94.38	61.31	86.78	62	86.22	45.26	90.76
	GReg	54.5	85.53	57.33	89.76	74.36	79.53	43.86	91.73	58.33	86.8	61.16	86.03	58.26	86.56
	GReg+	51.6	90	25.73	95.96	65.16	87.56	41.68	93.42	51.61	89.96	52.93	89.24	48.12	91.02
DenseNet	MSP [17]	86.12	70.66	85.55	72.33	83.25	74.01	77.45	78	92.18	57.09	90.71	60.59	85.87	68.78
	ODIN [28]	80.14	73.74	80.63	83.75	76.78	78.75	43.87	91.25	69.6	80.44	68.8	81.4	69.97	81.55
	Energy score [30]	84.81	70.29	88.6	80.99	78.12	77.48	51.08	88.73	84.56	69.31	86.3	69.96	78.91	76.12
	ReAct [46]	78.22	77.9	88.02	78.76	81.98	73.53	53.51	88.13	76.23	81.32	79.7	80.97	76.27	80.10
	DICE [47]	84.34	71.02	87.51	81.86	78.19	78.15	14.75	97.44	75.36	77.77	78.7	76.8	69.80	80.50
	LINe [1]	39.24	87.91	31.6	91.7	88.48	63.82	5.75	98.85	23.33	94.95	22.6	95.13	35.16	88.72
	OE [18]	74.55	77.07	61.17	87.34	66.75	81.56	24.98	95.1	65.02	83.12	70.65	81.11	60.52	84.21
	Energy Loss [30]	70.03	81.3	70.78	88.42	62.45	85.15	26.09	95.38	77.59	75.23	79.32	77.69	64.37	83.86
	POEM [35]	75.48	73.04	83.57	71.49	83.6	73.99	33.96	93.43	39.74	83.58	39.61	83.33	59.33	79.81
	OpenMix [55]	63.66	84.05	72.27	85.77	73.17	80.19	48.79	90.8	71.76	83.77	74.1	82.23	67.29	84.47
	DOS [22]	38.3	91.72	11.57	97.88	57.06	88.91	25.32	95.58	38.43	92.75	38.85	92.49	34.92	93.22
	GReg	64.63	81.74	45.79	92.53	78.1	78.1	31.27	94.53	55.98	89.43	61.96	87.8	56.29	87.35
	GReg+	41.83	90.19	14.44	97.06	53.56	89.11	19.49	95.8	26.01	94.36	27.95	93.77	30.55	93.38

Table 10: Comparison of methods with sampling on the CIFAR-10 benchmark. The experimental results are reported over three trials. AUROC and FPR95 are percentages.

Network	Method	OOD Datasets												Average	
		Textures		SVHN		Places		LSUN-c		LSUN-r		iSUN			
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
ResNet	POEM [35]	33.24	93.51	58.95	89.58	61.82	86.26	62.78	90.13	11.22	97.96	10.53	98.03	39.76	92.58
	DOS [22]	7.76	98.74	4.82	99.19	21.3	95.50	9.68	98.39	10.86	97.93	12.31	97.68	11.12	97.91
	GReg	5.96	98.36	3	98.83	22.76	94.8	3.03	99	6.5	98.4	6.2	98.33	**7.91**	**97.95**
	GReg+	7.70	98.47	5.65	98.79	24.25	94.80	9.54	98.18	14.56	97.30	15.04	97.11	12.79	97.44
WRN	POEM [35]	31.08	93.84	64.09	87.23	57.86	87.62	29.51	95.01	25.97	95.08	23.17	95.48	38.61	92.38
	DOS [22]	4.33	99	1.78	99.32	10.89	97.16	3.73	98.98	9.18	98.2	9.86	98.17	6.63	**98.47**
	GReg	7.83	98.16	6.23	98.4	19.9	95.66	3.8	98.93	5.26	98.7	4.66	98.76	7.95	98.10
	GReg+	4.19	98.52	2.92	98.59	11.38	96.71	3.88	98.63	5.63	98.47	6.57	98.27	**5.76**	98.2
DenseNet	POEM [35]	36.79	91.31	31.69	94.34	53.69	88.34	43.57	92.84	5.06	99.03	4.11	99.23	29.15	94.18
	DOS [22]	2.61	99.39	0.67	99.65	7.16	97.90	0.88	99.56	1.61	99.21	1.53	99.27	**2.41**	**99.16**
	GReg	8.3	97.9	3.7	98.83	20.13	95.66	3.1	99.23	6.36	98.53	6.03	98.56	7.93	98.12
	GReg+	6.21	98.29	2.19	98.73	21.5	95.1	6.87	98.27	15.53	97.45	15.36	97.43	11.27	97.54