Provable Guarantees for Understanding Out-of-Distribution Detection

Peyman Morteza, Yixuan Li

University of Wisconsin-Madison
{peyman, sharonli}@cs.wisc.edu

Abstract

Out-of-distribution (OOD) detection is important for deploying machine learning models in the real world, where test data from shifted distributions can naturally arise. While a plethora of algorithmic approaches have recently emerged for OOD detection, a critical gap remains in theoretical understanding. In this work, we develop an analytical framework that characterizes and unifies the theoretical understanding for OOD detection. Our analytical framework motivates a novel OOD detection method for neural networks, *GEM*, which demonstrates both theoretical and empirical superiority. In particular, on CIFAR-100 as in-distribution data, our method outperforms a competitive baseline by 16.57% (FPR95). Lastly, we formally provide provable guarantees and comprehensive analysis of our method, underpinning how various properties of data distribution affect the performance of OOD detection.

1 Introduction

When deploying machine learning models in the open world, it becomes increasingly critical to ensure the reliability—models are not only accurate on their familiar data distribution, but also aware of unknown inputs outside the training data distribution. Out-of-distribution (OOD) samples can naturally arise from an irrelevant distribution whose label set has no intersection with training categories, and therefore should not be predicted by the model. This gives rise to the importance of OOD detection, which determines whether an input is in-distribution (ID) or OOD.

The main challenge in OOD detection stems from the fact that modern deep neural networks can easily produce overconfident predictions on OOD inputs (Nguyen, Yosinski, and Clune 2015). This phenomenon makes the separation between ID and OOD data a non-trivial task. OOD detection approaches commonly rely on an OOD scoring function that derives statistics from the pre-trained neural networks and performs OOD detection by exercising a threshold comparison. For example, (Hendrycks and Gimpel 2017) use the maximum softmax probability (MSP) and classifies inputs with smaller MSP scores as OOD data. While improved OOD scoring functions (Liang, Li, and Srikant 2018; Lee et al. 2018b; Liu et al. 2020; Sun, Guo, and Li 2021) have emerged recently, their inherent connections and theoretical understandings are largely lacking. To the best of our knowledge, there is limited prior work providing provable guarantees for OOD detection methods from a rigorous mathematical point of view.

This paper takes an important step to bridge the gap by providing a unified framework that allows the research community to understand the theoretical connections among recent model-based OOD detection methods. Our framework further enables devising new methodology, theoretical and empirical insights on OOD detection. Our **key contributions** are three folds:

- First, we provide an analytical framework that precisely characterizes and unifies the theoretical interpretation of several representative OOD scoring functions (Section 2). We derive analytically an optimal form of OOD scoring function called *GEM (Gaussian mixture based Energy Measurement)*, which is provably aligned with the true log-likelihood for capturing OOD uncertainty. In contrast, we show mathematically that prior scoring functions can be sub-optimal.
- Second, our analytical framework motivates a new OOD detection method for deep neural networks (Section 3). By modeling the feature space as a class-conditional multivariate Gaussian distribution, we propose a *GEM* score based on the Gaussian generative model. Empirical evaluations demonstrate the competitive performance of the new scoring function. In particular, on CIFAR-100 as in-distribution data, *GEM* outperforms (Liu et al. 2020) by 16.57% (FPR95). Our method is theoretically more rigorous than maximum Mahalanobis distance (Lee et al. 2018b) while achieving equally strong performance.
- Lastly, our work provides both provable guarantees and empirical analysis to understand how various properties of data representation in feature and input space affect the performance of OOD detection (Section 4). Previous OOD detection methods can be difficult to analyze due to the stochasticity in neural network optimization. Our framework offers key simplifications that allow us to (1) isolate the effect of data representation from model optimization, and (2) flexibly modulate properties of data representation in feature and input space. Through both synthetic simulations and theoretical analysis, our study

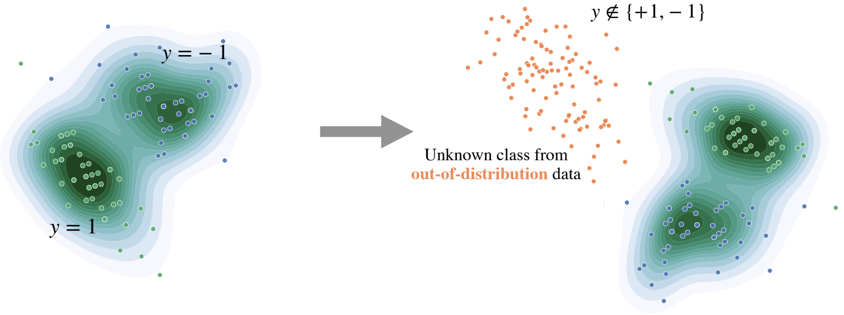

Figure 1: Left: The in-distribution data $P_{\mathcal{X}}^{\text{in}}$ comprises of two classes $\mathcal{Y} = \{-1, +1\}$, indicated by green and blue dots respectively. Right: Out-of-distribution detection allows the learner to express ignorance outside the support set of current known classes, and prevents the model from misclassifying OOD data (orange dots) into known classes (blue and green dots).

reveals important insights on how OOD detection performance changes with respect to data distributions.

We end the introduction with an outline of this work. In Section 2, we first define the problem of study and set the notations that we need. Next, we analyze previous OOD detection methods under the Gaussian mixture assumption and introduce the GEM score. In Section 3, we extend GEM to deep neural networks and perform experiments on common benchmarks. In Section 4, we provide rigorous guarantees for the performance of GEM, along with simulation verifications. We conclude our work in Section 6, following an expansive literature review in Section 5.

2 OOD Detection Under Gaussian Mixtures

In this section, we mathematically describe representative OOD scoring functions under the Gaussian mixture data model. This allows us to contrast with the ideal OOD detector where the data density is explicit. We later apply the insight gained from this simple model to introduce a new score OOD detection for deep neural networks.

Preliminaries

We denote by $\mathcal{X} = \mathbb{R}^d$ the input space and $\mathcal{Y} = \{y_1, ..., y_k\}$ the label space. Let $P_{\mathcal{X},\mathcal{Y}}^{\text{in}}$ denote a probability distribution defined on $\mathcal{X} \times \mathcal{Y}$. Furthermore, let $P_{\mathcal{X}}^{\text{in}}$ and $P_{\mathcal{Y}}^{\text{in}}$ denote the marginal probability distribution on $\mathcal{X}$ and $\mathcal{Y}$ respectively. A classifier $f : \mathcal{X} \to \mathbb{R}^k$ learns to map a given input $\mathbf{x} \in \mathcal{X}$ to the output space .

Problem Statement Given a classifier f learned on training samples from in-distribution $P_{\mathcal{X},\mathcal{Y}}^{\text{in}}$, the goal is to design a binary function estimator,

$$g : \mathcal{X} \to \{\text{in}, \text{out}\},$$

that classifies whether a test-time sample $\mathbf{x} \in \mathcal{X}$ is generated from $P_{\mathcal{X}}^{\text{in}}$ or not. Estimating OOD uncertainty is challenging due to the lack of knowledge on OOD data coming from $P_{\mathcal{X}}^{\text{out}}$. It is infeasible to explicitly train a binary classifier g. A natural approach is to use level set for OOD detection, based on the data density $P_{\mathcal{X}}^{\text{in}}$. We define the *ideal classifier* for OOD detection as follows,

$$g_{\lambda}^{\text{ideal}}(\mathbf{x}) = \begin{cases} \text{in} & p_{\mathcal{X}}^{\text{in}}(\mathbf{x}) \geq \lambda \\ \text{out} & p_{\mathcal{X}}^{\text{in}}(\mathbf{x}) < \lambda \end{cases},$$

where $p_{\mathcal{X}}^{\text{in}}$ is the density function of $P_{\mathcal{X}}^{\text{in}}$ and λ is the threshold, which is chosen so that a high fraction (*e.g.,* 95%) of in-distribution data is correctly classified. For evaluation purpose, we define the error rate by,

$$\text{TPR}(g) := \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}^{\text{in}}}(\mathbb{I}_{\{g(\mathbf{x})=\text{in}\}}),$$
$$\text{FPR}(g) := \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}^{\text{out}}}(\mathbb{I}_{\{g(\mathbf{x})=\text{in}\}}).$$

By convention, we assume in-distribution samples have positive labels. In practice, $P_{\mathcal{X}}^{\text{out}}$ is often defined by a distribution that simulates unknowns encountered during deployment time, such as samples from an irrelevant distribution whose label set has no intersection with $\mathcal{Y}$ and therefore should not be predicted by the model.

In-distribution Data Model We assume in-distribution data is drawn from a Gaussian mixture with equal priors and a tied covariance matrix Σ. The simplicity is desirable for us to precisely characterize various OOD detection methods and their optimality. We will further extend our analysis to neural networks in Section 3. Specifically,

$$\mathbf{x}|y_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma),$$
$$p_{\mathcal{Y}}^{\text{in}}(y_i) = \frac{1}{k},$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^d$ is the mean of class $y_i \in \mathcal{Y}$ and $\Sigma \in \mathbb{R}^{d \times d}$ is the covariance matrix. The class-conditional density follows a Gaussian distribution,

$$p_{\mathcal{X}|\mathcal{Y}}^{\text{in}}(\mathbf{x}|y_i) = \frac{\exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_i))}{\sqrt{(2\pi)^d |\Sigma|}}.$$

Above implies the density function of $P_{\mathcal{X}}^{\text{in}}$ can be written as follows,

$$\begin{aligned} p_{\mathcal{X}}^{\text{in}}(\mathbf{x}) &= \sum_{j=1}^{k} p_{\mathcal{X}|\mathcal{Y}}^{\text{in}}(\mathbf{x}|y_j) \cdot p_{\mathcal{Y}}^{\text{in}}(y_j) \\ &= \frac{\sum_{j=1}^{k} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_j))}{k\sqrt{(2\pi)^d |\Sigma|}}, \end{aligned}$$

which is mixture of k Gaussian distributions.

Bayes Optimal Classifier Under the Gaussian mixture model, the posterior probability of a Bayes optimal classifier for class $y_i \in \mathcal{Y}$ is given by,

$$p_{\mathcal{Y}|\mathcal{X}}(y_i|\mathbf{x}) = \frac{p_{\mathcal{Y}}(y_i)p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|y_i)}{\sum_{j=1}^{k} p_{\mathcal{Y}}(y_j)p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|y_j)} \tag{1}$$

$$= \frac{\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i))}{\sum_{j=1}^{k}\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_j))} \tag{2}$$

$$= \frac{\exp f_i(\mathbf{x})}{\sum_{j=1}^{k}\exp f_j(\mathbf{x})}, \tag{3}$$

where $f : \mathcal{X} \to \mathbb{R}^k$ is a function mapping to the logits. One can note that the above form of posterior distribution is equivalent to applying the softmax function on the logits $f(\mathbf{x})$, where,

$$f_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i),$$

which is also known as the Mahalanobis distance (Mahalanobis 1936).

OOD Scoring Functions and Their Optimality

We now contrast several representative OOD scoring functions and also introduce our new scoring function GEM. Note that an ideal OOD detector should use a scoring function that is proportional to the data density. We focus on post hoc OOD detection methods, which have the advantages of being easy to use and general applicability without modifying the training procedure and objective.

Prior: Maximum Softmax Score Hendrycks and Gimpel propose using the maximum softmax score (MSP) for estimating OOD uncertainty,

$$g_\lambda^{\text{MSP}}(\mathbf{x}) = \begin{cases} \text{in} & \text{MSP}(f,\mathbf{x}) \geq \lambda \\ \text{out} & \text{MSP}(f,\mathbf{x}) < \lambda \end{cases}.$$

The OOD scoring function is given by,

$$\begin{aligned} \text{MSP}(f,\mathbf{x}) &= \max_i p_{\mathcal{Y}|\mathcal{X}}(y_i|\mathbf{x}) \\ &= \max_i \frac{1}{k\beta p_{\mathcal{X}}^{\text{in}}(\mathbf{x})}\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)) \\ &\not\propto p_{\mathcal{X}}^{\text{in}}(\mathbf{x}), \end{aligned}$$

where $\beta = \sqrt{(2\pi)^d|\Sigma|}$. The above suggests that MSP is not aligned with the true data density, as illustrated in Figure 2. For simplicity, we visualize the case when the input distribution is mixture of one-dimensional Gaussians, with two classes $\mathcal{Y} = \{+1, -1\}$. MSP can yield high score 1, and misclassify data points in low-likelihood regions such as $x > 4$ or $x < -4$ (highlighted in red). Also, depending on threshold value λ, MSP may misclassify samples from neighbourhood around the origin (highlighted in gray).

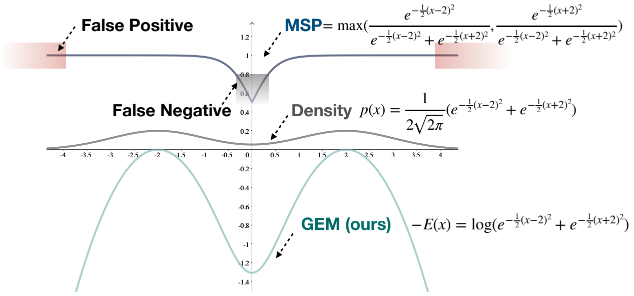

Figure 2: Illustration of data density (middle, gray), maximum softmax score (top, black) and GEM score (bottom, blue) when $x \in \mathbb{R}$. The data distribution $p(x)$ is a mixture of two Gaussians with mean $\mu_1 = 2$ and $\mu_2 = -2$ respectively. The variance $\sigma_1 = \sigma_2 = 1$. Under thresholding, MSP can not distinguish samples $x > 4$ or $x < -4$ which have low-likelihood of being in-distribution. In contrast, GEM score (ours) is aligned with the true data density, and better captures OOD uncertainty.

Prior: Maximum Mahalanobis Distance Lee et al. propose using the maximum Mahalanobis distance *w.r.t* the closest class centroid for OOD detection. Specifically, the score is defined as:

$$\begin{aligned} M(f,\mathbf{x}) &= \max_i -(\mathbf{x}-\boldsymbol{\mu}_i)^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) \\ &\not\propto p_{\mathcal{X}}^{\text{in}}(\mathbf{x}), \end{aligned}$$

which is equivalent to the maximum Mahalanobis distance. The corresponding OOD classifiers based on Mahalanobis score is,

$$g_\lambda^{\text{Mahalanobis}}(\mathbf{x}) = \begin{cases} \text{in} & M(f,\mathbf{x}) \geq \lambda \\ \text{out} & M(f,\mathbf{x}) < \lambda \end{cases}.$$

The above suggests that Mahalanobis distance is not proportional to the true data density either, hence sub-optimal.

Prior: Energy Score Given a function transformation $f : \mathcal{X} \to \mathbb{R}^k$, Liu et al. propose using the free energy score for OOD detection. The free energy is defined to be the `-logsumexp` of logit outputs,

$$\text{E}(f,\mathbf{x}) = -\log\sum_{j=1}^{k}\exp(f_j(\mathbf{x})), \tag{4}$$

where $f(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_k(\mathbf{x}))^\top \in \mathbb{R}^k$. We provide a simple and concrete example to show there exists maximum likelihood solution with the same posterior probability as in Equation 3, but the resulting energy score is not aligned with the data density:

$$\begin{aligned} p_{\mathcal{Y}|\mathcal{X}}(y_i|\mathbf{x}) &= \frac{\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i))}{\sum_{j=1}^{k}\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^\top\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_j))} \\ &= \frac{\exp(\boldsymbol{\mu}_i^\top\Sigma^{-1}\mathbf{x}-\frac{1}{2}\boldsymbol{\mu}_i^\top\Sigma^{-1}\boldsymbol{\mu}_i)}{\sum_{j=1}^{k}\exp(\boldsymbol{\mu}_j^\top\Sigma^{-1}\mathbf{x}-\frac{1}{2}\boldsymbol{\mu}_j^\top\Sigma^{-1}\boldsymbol{\mu}_j)} \\ &= \frac{\exp f_i'(\mathbf{x})}{\sum_{j=1}^{k}\exp f_j'(\mathbf{x})}, \end{aligned}$$

where $f'(\mathbf{x}) := (f'_1(\mathbf{x}), ..., f'_k(\mathbf{x}))^\top \in \mathbb{R}^k$ can be viewed as a single layer network's output with (row) weights $\boldsymbol{\mu}_i^\top \Sigma^{-1}$, for $1 \leq i \leq k$, and biases $-\frac{1}{2}\boldsymbol{\mu}_i^\top \Sigma^{-1}\boldsymbol{\mu}_i$, for $1 \leq i \leq k$, and the corresponding energy $-\log \sum_j \exp f'_j(\mathbf{x})$ is not aligned with the log-likelihood, hence not always optimal.

New: GEM Score We now introduce a new scoring function, Gaussian mixture based energy measurement (dubbed *GEM*). The GEM score can be written as,

$$\begin{aligned}\mathrm{GEM}(f, \mathbf{x}) &= -E(f, \mathbf{x}) \\ &= \log \sum_{j=1}^{k} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_j)) \\ &\propto \log p_{\mathcal{X}}^{\text{in}}(\mathbf{x}),\end{aligned}$$

which suggests that the *GEM score* is proportional (by ignoring a constant term) to the log-likelihood of the in-distribution data. Note that we flip the sign of free energy to align with the convention that larger GEM score indicates more ID-ness and vice versa. The key difference here is that the GEM score is a *special case* of negative free energy, where each $f_j(\mathbf{x})$ in Equation 4 takes on the form of Mahalanobis distance instead of directly using the logit outputs,

$$f_j(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_j).$$

In Figure 2, we show the alignment between the GEM (light green) and true data density function (gray), in a simplified case with $x \in \mathbb{R}$, $k = 2$ and $\mu_1 = 2, \mu_2 = -2$. The corresponding OOD classifiers based on energy score is,

$$g_\lambda^{\text{GEM}}(\mathbf{x}) = \begin{cases} \text{in} & \text{GEM}(f, \mathbf{x}) \geq \lambda \\ \text{out} & \text{GEM}(f, \mathbf{x}) < \lambda \end{cases}.$$

This leads to the following lemma that shows the optimality of the GEM estimator.

Lemma 1. *In the case of Gaussian conditional with equal priors, the GEM based OOD estimator performs similarly to the ideal classifier defined in Section 2. More specifically,*

$$g_\lambda^{ideal} = g_{\log(k\beta\cdot\lambda)}^{GEM},$$

where $\beta = \sqrt{(2\pi)^d|\Sigma|}$ and both the ideal classifier and our method are aligned with $P_{\mathcal{X}}^{\text{in}}$ by definition.

Remark 1. Note that the equal prior case is considered to convey the main idea in simplest possible form. To make it more general, a weighted version of GEM can be used to achieve the optimality for the non-equal prior case. More precisely, let $w_i = p_{\mathcal{Y}}^{\text{in}}(y_i)$, then we have,

$$\begin{aligned}p_{\mathcal{X}}^{\text{in}}(\mathbf{x}) &= \sum_{j=1}^{k} w_j p_{\mathcal{X}|\mathcal{Y}}^{\text{in}}(\mathbf{x}|y_j) \\ &\propto \sum_{j=1}^{k} w_j \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_j)).\end{aligned}$$

Now if we define the *weighted GEM* by,

$$\text{GEM}^w(f, \mathbf{x}) := \log \sum_{j=1}^{k} w_j \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_j)),$$

then arguing similar to Lemma 1 implies that weighted GEM would be aligned with the ideal classifier in the non-equal prior case.

3 OOD Detection for Deep Neural Networks

In this section, we extend our analysis and method to deep neural networks. To start, let $h(\mathbf{x}; \theta) \in \mathbb{R}^m$ be the feature vector of the input $\mathbf{x}$, extracted from the penultimate layer of a neural net parameterized by θ. We assume that a class-conditional distribution in the feature space follows the multivariate Gaussian distribution. Such an assumption has been empirically validated in (Lee et al. 2018b); also see visualizations in Figure 3. Specifically, a k class-conditional Gaussian distribution with a tied covariance is defined as,

$$h(\mathbf{x}; \theta)|y_i \sim \mathcal{N}(\boldsymbol{u}_i, \bar{\Sigma}),$$

where $\boldsymbol{u}_i \in \mathbb{R}^m$ is the mean of class y_i and $\bar{\Sigma} \in \mathbb{R}^{m\times m}$ is the covariance matrix. To estimate the parameters of the generative model from the pre-trained neural classifier, one can compute the empirical class mean and covariance given training samples $\{(\mathbf{x}_1, \bar{y}_1), (\mathbf{x}_2, \bar{y}_2), ..., (\mathbf{x}_N, \bar{y}_N)\}$,

$$\begin{aligned}\hat{\boldsymbol{u}}_i &= \frac{1}{N_i} \sum_{j:\bar{y}_j = y_i} h(\mathbf{x}_j; \theta), \\ \hat{\Sigma} &= \frac{1}{N} \sum_{i=1}^{k} \sum_{j:\bar{y}_j = y_i} (h(\mathbf{x}_j; \theta) - \hat{\boldsymbol{u}}_i)(h(\mathbf{x}_j; \theta) - \hat{\boldsymbol{u}}_i)^T,\end{aligned}$$

where N_i is the number of training samples with label $y_i \in \mathcal{Y}$. We can define the ideal classifier with respect to *feature space* to be,

$$g_\lambda^{\text{ideal}}(\mathbf{x}) = \begin{cases} \text{in} & p^{\text{feature}}(\mathbf{x}) \geq \lambda \\ \text{out} & p^{\text{feature}}(\mathbf{x}) < \lambda, \end{cases} \quad (5)$$

where p^{feature} denotes the density function of the posterior distribution on the feature space induced by $h(\mathbf{x}, \theta)$.

GEM for Neural Networks Similar to our definition in Section 2, *GEM* for neural networks can be defined as

$$\text{GEM}(\mathbf{x}; \theta) = \log \sum_{j=1}^{k} \exp(f_j(\mathbf{x}; \theta)),$$

where $f_j(\mathbf{x}; \theta) = -\frac{1}{2}(h(\mathbf{x}; \theta) - \boldsymbol{u}_j)^\top \bar{\Sigma}^{-1}(h(\mathbf{x}; \theta) - \boldsymbol{u}_j)$. We can empirically estimate each $f_j(\mathbf{x}; \theta)$ by,

$$\hat{f}_j(\mathbf{x}; \theta) = -\frac{1}{2}(h(\mathbf{x}; \theta) - \hat{\boldsymbol{u}}_j)^\top \hat{\Sigma}^{-1}(h(\mathbf{x}; \theta) - \hat{\boldsymbol{u}}_j).$$

It follows from an analogue of Lemma 1 that g_λ^{GEM}, computed from feature space, performs similarly to the ideal classifier that we defined by Equation 5.

Lemma 2. *The performance of GEM based detection is same as the ideal classifier (with respect to the feature space) defined by Equation 5 :*

$$g_\lambda^{ideal} = g_{\log(k\bar{\beta}\cdot\lambda)}^{GEM},$$

where $\bar{\beta} = \sqrt{(2\pi)^m|\bar{\Sigma}|}$.

We also note that Lemma 2 can be extended to non-equal prior case by arguing similar to Remark 1.

In-distribution	Method	FPR95 ↓	AUROC ↑	AUPR ↑	In-dist Test Error ↓
CIFAR-10	Softmax score (Hendrycks and Gimpel 2017)	51.04	90.90	97.92	5.16
	ODIN (Liang, Li, and Srikant 2018)	35.71	91.09	97.62	5.16
	Mahalanobis (Lee et al. 2018b)	36.96	93.24	98.47	5.16
	Energy score (Liu et al. 2020)	33.01	91.88	97.83	5.16
	GEM (ours)	37.21	93.23	98.47	5.16
CIFAR-100	Softmax score (Hendrycks and Gimpel 2017)	80.41	75.53	93.93	24.04
	ODIN (Liang, Li, and Srikant 2018)	74.64	77.43	94.23	24.04
	Mahalanobis (Lee et al. 2018b)	57.01	82.70	95.68	24.04
	Energy score (Liu et al. 2020)	73.60	79.56	94.87	24.04
	GEM (ours)	57.03	82.67	95.66	24.04

Table 1: Main Results. Comparison with competitive *post hoc* OOD detection methods. ↑ indicates larger values are better, and ↓ indicates smaller values are better. All values are percentages. Results for OOD detection are averaged over the six OOD test datasets described in section 3. Numbers for individual OOD test datasets are available in the extended version (Morteza and Li 2021). The reported results for baselines are courtesy of (Liu et al. 2020).

Experimental Results

Setup We use CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009) datasets as in-distribution data.[1]. We use the standard split, and train with WideResNet architecture (Zagoruyko and Komodakis 2016) with depth 40. For the OOD test dataset, we use the following six datasets: `Textures` (Cimpoi et al. 2014), `SVHN` (Netzer et al. 2011), `Places365` (Zhou et al. 2017), `LSUN-Crop` (Yu et al. 2015), `LSUN-Resize` (Yu et al. 2015), and `iSUN` (Xu et al. 2015). We report standard metrics including FPR95 (false positive rate of OOD examples when the true positive rate of in-distribution examples is at 95%), AUROC, and AUPR.

GEM is both empirically competitive and theoretically grounded. Table 1 compares the performance of the GEM method with common OOD detection methods. For fairness, all methods derive OOD scoring functions post hoc from the same pre-trained model. For example, on CIFAR-100 as in-distribution data, GEM outperforms the energy score (Liu et al. 2020) by 16.57% (FPR95). Compared to (Lee et al. 2018b), our method is more theoretically grounded than taking the maximum Mahalanobis distance. We note that the similar empirical performance is primarily due to `log-sum-exp` being a smooth approximation of maximum Mahalanobis distance in the feature space (more details in Remark 2 below). Therefore, our method overall achieves both strong empirical performance and theoretical soundness—bridging a critical gap under unified understandings.

Remark 2 (Significance *w.r.t* Mahalanobis)**.** The main difference *w.r.t* (Lee et al. 2018b) is that we are taking the `log-sum-exp` over Mahalanobis distances M_i, instead of taking the `maximum` Mahalanobis distance. This was motivated by our theoretical analysis in previous Section where taking `log-sum-exp` would be aligned with likelihood (*w.r.t* feature space), whereas `max` is not exact in theory. In other words, we bring theoretical rigor to an empirically competitive method. Mathematically, $\log \sum_i^k \exp(M_i) \approx \max_i M_i$ with the following bound: $\max_i M_i \leq \log \sum_i^k \exp(M_i) \leq \max_i M_i + \log(k)$. Therefore, our method overall achieves equally strong empirical performance yet with theoretical soundness and guarantees (see formal analysis in Section 4).

Lemma 3. *In the case of Gaussian conditional with equal priors in the feature space, the Mahalanobis-based OOD estimator is not aligned with the density of in-distribution data in the feature space and it is not equivalent to the ideal classifier defined by Equation 5.*

Remark 3 (Significance *w.r.t* Energy Score)**.** The energy score in (Liu et al. 2020) was derived directly from the logit outputs, rather than a Gaussian generative model as in ours. As a result, the original energy score might not always correspond to the Bayes optimal logit to ensure alignment *w.r.t* likelihood (we showed this by an explicit example in Section 2). Instead, our analytical framework and method provide strong provable guarantees (c.f. Section 4) and enable precise understanding by disentangling the effects of various factors (c.f. Section 4), both of which were not presented in (Liu et al. 2020). Moreover, we show empirically that GEM achieves strong empirical performance, outperforming energy score by a significant margin (16.57% in FPR95 on CIFAR-100, see Table 1).

4 Provable Guarantees for GEM

The main goal of this section is to provide rigorous guarantees and understandings for our method GEM. This is important but often missing in previous literature on OOD detection.

Let $P_{\mathcal{X}}^{\text{in}}$ be a mixture of Gaussians (similar to Section 2) and assume $P_{\mathcal{X}}^{\text{out}} = \mathcal{N}(\boldsymbol{\mu}_{\text{out}}, \Sigma)$. We can think of $\mathcal{X}$ as either the feature space or input space of a deep neural net. We work with the re-scaled version of the GEM score (by omitting the log operator), which does not change the formal guarantees.

$$ES(\mathbf{x}) = \sum_{i=1}^{k} ES_i(\mathbf{x}),$$

[1] Code is available at: https://github.com/PeymanMorteza/GEM

where,

$$ES_i(\mathbf{x}) = \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)).$$

Next, we consider the following quantity,

$$D := \mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}^{\text{in}}}(ES(\mathbf{x})) - \mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}^{\text{out}}}(ES(\mathbf{x})).$$

Intuitively, we can think of D as a measure of how well GEM distinguishes ID samples from OOD samples. For example, when $\boldsymbol{\mu}_{\text{out}}$ is far away from $\boldsymbol{\mu}_i$ then we expect $\mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}^{\text{out}}}(ES(\mathbf{x}))$ to be small (i.e., D is large), and we expect that the our OOD estimator performs better compared to the case when $\boldsymbol{\mu}_{\text{out}}$ is close to $\boldsymbol{\mu}_i$ (i.e., D is small). We make this intuition precise by bounding D in terms of Mahalanobis distance between $\boldsymbol{\mu}_{\text{out}}$ and $\boldsymbol{\mu}_i$. First, we recall the following definition and set some notations,

Definition 1. *For $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, the Mahalanobis distance, with respect to Σ, is defined by,*

$$d_M(\mathbf{u}, \mathbf{v}) := \sqrt{(\mathbf{u}-\mathbf{v})^\top \Sigma^{-1}(\mathbf{u}-\mathbf{v})},$$

and for $r > 0$ the open ball with center $\mathbf{u}$ and radius r is defined by,

$$B_r(\mathbf{u}) := \{\mathbf{x} \in \mathbb{R}^d | d_M(\mathbf{x}, \mathbf{u}) < r\}.$$

Next, we can state the following theorem.

Theorem 1. *We have the following bounds,*

- $\mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}^{\text{out}}}(ES(\mathbf{x})) \leq \sum_{i=1}^{k}\Big(\big(1 - P_{\mathcal{X}}^{\text{out}}(B_{\alpha_i}(\boldsymbol{\mu}_{\text{out}}))\big) + \exp(-\frac{1}{2}\alpha_i^2)\Big),$
- $\mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}^{\text{in}}}(ES(\mathbf{x})) - \mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}^{\text{out}}}(ES(\mathbf{x})) \leq \sum_{i=1}^{k}\alpha_i,$

where, for $1 \leq i \leq k$, $\alpha_i := \frac{1}{2}d_M(\boldsymbol{\mu}_i, \boldsymbol{\mu}_{\text{out}})$.

We emphasize that in Theorem 1 $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_{\text{out}}$ can have *arbitrary configurations*. We refer the reader to the extended version[2] (Morteza and Li 2021) for the proof of Theorem 1 and detailed discussions on other variants.

Performance with respect to the distance between ID and OOD data The next corollary explains how Theorem 1 can quantify that the performance of GEM-based OOD detector increases as the distance between ID and OOD data increases.

Corollary 1. *For $1 \leq i \leq k$, set $\alpha = d_M(\boldsymbol{\mu}_{out}, \boldsymbol{\mu}_i)$. We have the following from the first bound in Theorem 1,*

$$\mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}^{\text{out}}}(ES(\mathbf{x})) \leq k\Big(\big(1 - P_{\mathcal{X}}^{\text{out}}(B_{\alpha}(\boldsymbol{\mu}_{\text{out}}))\big) + \exp(-\frac{1}{2}\alpha^2)\Big).$$

Now as $\alpha \to \infty$ the right hand side in the above approaches to 0. This indicates that the performance of our method improves as $\alpha \to \infty$. On the other hand, using the second bound in the Theorem 1, we have,

$$\mathbb{E}_{x\sim P_{\mathcal{X}}^{\text{in}}}(ES(\mathbf{x})) - \mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}^{\text{out}}}(ES(\mathbf{x})) \leq k\alpha,$$

and it follows that as $\alpha \to 0$ the energy difference between in-distribution and out-of-distribution data converges to 0. In other words, the performance decreases as α approaches to 0. We will further justify our theory in simulation study (next subsection).

Performance in high dimensions We now show that the performance of GEM decreases as dimension of feature space increases. This is due to *curse of dimensionality* which we next explain. First, for simplicity assume that $\boldsymbol{\mu}_{out} = 0$ and for all $1 \leq i \leq k$, $\alpha = d_M(\boldsymbol{\mu}_{out}, \boldsymbol{\mu}_i)$. Consider a multi-dimensional gaussian $\mathcal{N}(0, \mathbf{I}_d)$. As d increases the high-probability region under this gaussian distribution will concentrate away from the origin. More precisely,

$$\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}_d) \implies \|\mathbf{x}\|_2^2 \sim \chi_d^2 \implies \mathbb{E}(\|\mathbf{x}\|_2^2) = d.$$

Therefore, the out-of-distribution samples will have a larger distance (on average) to the origin as dimension increases and it follows that the OOD detector may misclassify these OOD samples as in-distribution.

We next conduct several simulation studies to systematically verify our provable guarantees.

Simulation Studies and Further Analysis

What properties of the data representation make OOD uncertainty challenging? In this subsection, we construct a synthetic data representation that allows us to flexibly modulate different properties of the data representation including:

(i) distance between ID and OOD data,

(ii) feature or input dimension,

(iii) number of classes.

We simulate and probe how these factors affect OOD uncertainty estimation. The simulation also serves as a verification of our theoretical guarantees.

Feature representation setup The in-distribution representation on the feature space (or input space) comprises a mixture of k class-conditional Gaussian. To replicate common empirical benchmarks such as CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009), we explore both $k = 10$ and $k = 100$ by default. Unless otherwise specified, we set the feature (or input) dimension $d = 512$. We fix the total number of in-distribution samples $N = 20,000$. The tied covariance matrix is diagonal with magnitude σ^2, i.e., $\Sigma = \sigma^2\mathbf{I}_d$.

We assume the data in the feature space (or input space) $\mathbf{x} \in \mathbb{R}^d$ is sampled from the following class-conditional Gaussian,

$$\mathbf{x}_{\text{in}} \mid y_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2\mathbf{I}_d),$$

where $\boldsymbol{\mu}_i$ is the mean for in-distribution classes $i \in \{1, 2, ..., k\}$. We consider different configurations of $\boldsymbol{\mu}_i$, $1 \leq i \leq k$ representing means of each k in-distribution classes. Specifically, the mean $\boldsymbol{\mu}_i$ corresponding to i-th class is a unit vector $\boldsymbol{v}_i$, scaled by a distance parameter $r > 0$. In particular, $\boldsymbol{\mu}_i = r \cdot \boldsymbol{\nu}_i$, where $\|\boldsymbol{\nu}_i\|_2 = 1$. $\boldsymbol{\nu}_i$ is a sparse vector with $s = \lfloor d/k \rfloor$ non-zero entries, with equal values in the position from $(i-1)\cdot s$ up to $i \cdot s$ and 0 elsewhere. It follows that for $i, j \in \{1, ..., k\}$ and $i \neq j$,

$$\begin{aligned} &\langle \boldsymbol{\nu}_i, \boldsymbol{\nu}_j \rangle = 0, \\ &\|\boldsymbol{\nu}_i - \boldsymbol{\nu}_j\|_2 = \sqrt{2}. \end{aligned} \tag{6}$$

[2] Available at: https://arxiv.org/abs/2112.00787

Furthermore, we assume that the out-of-distribution data representation is centered at the origin, with $\boldsymbol{\mu}_{\text{out}} = \mathbf{0} \in \mathbb{R}^d$,

$$\mathbf{x}_{\text{out}} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d).$$

Note that the above configuration is considered for simplicity and similar simulation results holds when we translate $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_{\text{out}}$ with a constant vector or by applying an orthogonal transformation.

Rationale of the synthetic data Compared to estimating GEM scores from real datasets using parameterized models (such as neural networks), these synthetic simulations offer two key simplifications. First, viewing the setting on feature space, we can *flexibly modulate* key properties of datasets such as the number of classes and distance between induced ID and OOD representation in the feature space. In contrast, in real datasets, these properties are usually predetermined. Second, viewing the setting on input space, the function mapping $f(\mathbf{x})$ is completely deterministic and optimal, provided with known parameters $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k\}$ and Σ. This allows us to isolate the effect of data distribution from model optimization. In contrast, estimating $f(\mathbf{x})$ using complex models such as neural networks might have inductive bias, and depend on the optimization algorithm chosen.

Performance with respect to the number of classes We now show that the performance of our method decreases as the number of classes increases. To explain this, we compute D in terms of k to see how they are related. First, we need the following definition,

Definition 2. *Let $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^d$ with $\gamma = \|\boldsymbol{\mu} - \boldsymbol{\nu}\|_2$. Let $P \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_d)$. Define,*

$$A_\gamma := \mathbb{E}_{\mathbf{x} \sim P}(\exp(-\frac{1}{2}\|\mathbf{x} - \boldsymbol{\nu}\|_2^2)).$$

Remark 4. *Notice that, since standard Gaussian distribution is rotationally invariant, A_γ only depends on the distance between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ (i.e. γ). Also it is easy to see that A_γ decreases as γ increases.*

Proposition 2. *We have the following,*

$$D = A_0 - A_r + (k-1)(A_{\sqrt{2}\cdot r} - A_r).$$

We refer the reader to the extended version (Morteza and Li 2021) for the proof of Proposition 2. The next Corollary explains how the performance of our method decreases by increasing the number of classes.

Corollary 2. *Since $\sqrt{2} \cdot r > r$, it follows from Remark 4 that $A_{\sqrt{2}\cdot r} < A_r$. This means that the last term in the following is negative,*

$$D = A_0 - A_r + (k-1)(A_{\sqrt{2}\cdot r} - A_r).$$

In other words, as k increases D becomes smaller which indicates that the performance of the GEM method decreases.

Simulation Results

In this subsection, we report simulation results that confirm our theoretical guarantees presented above.

Effect of distance between ID and OOD Figure 3 (left) shows how the False Positive Rate (at 95% TPR) changes with the distance between ID and OOD features. The σ is set to be 1 and the distance is modulated by adjusting the magnitude parameter r, where a larger r results in a larger distance. For both $k = 10$ and $k = 100$, the FPR decreases as the distance increases, which matches our intuition that more drastic distribution shifts are easier to be detected. Under the same distance, we observe a relatively higher FPR for data with more classes ($k = 100$). The performance gap diminishes as the distance becomes very large.

Higher dimension exacerbates OOD uncertainty Figure 3 (middle) shows how the FPR changes as we increase the input dimension from $d = 100$ to $d = 1,000$ while keeping the distance fixed with $r = 10$ and $\sigma = 1$. As the dimension d increases, the number of non-zero entries in each $\boldsymbol{\mu}_i$ increases accordingly (i.e. $\boldsymbol{\mu}_i$ becomes less sparse). Under the same feature dimension, we observe a higher FPR for $k = 100$ than $k = 10$, which corroborates the empirical observations on CIFAR-10 and CIFAR-100 (Section 3). This suggests that higher dimensions can be a key factor inducing the detrimental effect in OOD detection.

Effect of the number of classes Lastly, we investigate the performance of OOD uncertainty estimation by linearly increasing the number of classes k from 10 to 100. We keep the magnitude parameter fixed with $r = 10$ and dimension $d = 512$ and $\sigma = 1$. We see as the number of classes increases, the performance of our method decreases. We close this section by noting that we also provided formal mathematical justifications in the previous subsection.

5 Related Work

Detecting unknowns has a long history in machine learning. We review works that are studied this problem in the context of deep neural networks. See (Yang et al. 2021) for a survey on generalized OOD detection (an umbrella term that includes closely related domains such as anomaly detection, novelty detection, open-set recognition, and OOD detection).

Out-of-distribution detection for discriminative models In (Bendale and Boult 2015), the OpenMax score is developed for OOD detection based on the extreme value theory (EVT). Subsequent work by Hendrycks and Gimpel proposed a simple baseline using maximum softmax probability. The MSP score for OOD input is proven to be arbitrarily high for neural networks with ReLU activation (Hein, Andriushchenko, and Bitterwolf 2019). Liang, Li, and Srikant improved MSP by proposing the ODIN score, which amplifies the ID and OOD separability. It is shown that a sufficiently large temperature has a strong smoothing effect that transforms the softmax score back to the logit space—which more effectively distinguishes between ID vs. OOD. In (Lee et al. 2018b), a score is constructed based on the maximum Mahalanobis distance to the class means in the feature space of the pre-trained network. Liu et al., proposed using the energy score, which can be derived directly from the logit output of the pre-trained network. In (Huang and Li 2021),

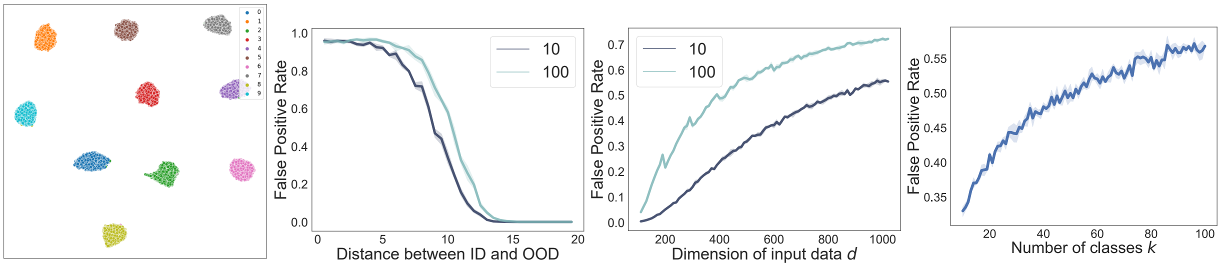

Figure 3: Left: UMAP visualization of embeddings for CIFAR-10 model. Performance of our method under induced configurations in feature space, including the distance between ID and OOD data (middle left), the dimension of input data d (middle right), and number of classes (right). Each curve is averaged over 5 different runs (shade indicates the variance). A lower value on the y-axis is better.

OOD detection is studied when the label space is large. It is shown that grouping the labels for in-distribution data can be effective in OOD detection for large semantic space. In (Ming, Yin, and Li 2022), the effect of spurious correlation is studied for OOD detection. Huang, Geng, and Li derived a scoring function termed GradNorm from the gradient space. GradNorm employs the vector norm of gradients, backpropagated from the KL divergence between the softmax output and a uniform probability distribution. In (Wang et al. 2021), the OOD detection is studied for multi-label classification where each data instance has multiple labels. In this work, we develop an analytical framework to analyze the performance of OOD scoring functions and show the superiority of GEM both theoretically and empirically.

Out-of-distribution detection via generative modeling There are several works that attempt modeling OOD data using generative modeling (e.g. GANs). Lee et al. use GANs to generate data with low density for model regularization. Vernekar et al. model in-distribution as a low dimensional submanifold of input space and uses auto-encoders to generate OOD samples outside of the in-distribution domain. Sricharan and Srivastava use GANs to generate OOD samples that the initial classifier is confident about and use those to create a more robust OOD detector. Prior research also used generative modeling to estimate the density of the in-distribution data, and classify a sample as OOD if the estimated likelihood is low. However, it is shown in (Nalisnick et al. 2019) that deep generative models can produce a higher likelihood for OOD data. For example, it fails to distinguish CIFAR10 samples from SVHN. In (Ren et al. 2019) and (Serrà et al. 2020), this problem is addressed by considering a likelihood ratio and taking the input complexity into account.

Out-of-distribution detection by model regularization Several works address the out-of-distribution detection problem during training-time regularization (Lee et al. 2018a; Bevandić et al. 2018; Geifman and El-Yaniv 2019; Malinin and Gales 2018a; Mohseni et al. 2020; Jeong and Kim 2020; Chen et al. 2021). In (Lee et al. 2018a), a new term is added to the loss function of the neural net to force the out-of-distribution sample to have uniform prediction values across labels. A similar loss is followed by outlier exposure (Hendrycks, Mazeika, and Dietterich 2018). In (Liu et al. 2020; Du et al. 2022), a term is added to the loss function of the network to force out-distribution samples to have higher energy values after training. In (Chen et al. 2021), an informative outlier mining procedure is proposed, which adaptively samples from auxiliary OOD data that is near the decision boundary between ID and OOD.

Such methods typically require having access to auxiliary unlabeled data. We focus on post hoc OOD detection methods, which have the advantages of being easy to use and general applicability. This is convenient for the adoption of OOD detection methods in real-world production environments, where the overhead cost of retraining or modifying the model can be prohibitive.

Uncertainty estimation in deep neural networks A Bayesian model is a statistical model that implements Bayes' rule to infer uncertainty within the model (Jaynes 1986). Recent works attempt several approximations of Bayesian inference including MC-dropout (Gal and Ghahramani 2016) and deep ensembles (Dietterich 2000; Lakshminarayanan, Pritzel, and Blundell 2017). These methods address model uncertainty (*i.e.*, epistemic) and are less competitive for OOD uncertainty estimation. Kendall and Gal developed an extended framework to study aleatoric and epistemic uncertainty together. In (Van Amersfoort et al. 2020) an uncertainty estimation method is developed using the RBF network. Dirichlet Prior Network (DPN) is also used for OOD detection with an uncertainty modeling of three different sources of uncertainty: model uncertainty, data uncertainty, and distributional uncertainty and form a line of works (Malinin and Gales 2018b, 2019; Nandy, Hsu, and Lee 2020).

6 Conclusion

In this work, we develop an analytical framework that precisely characterizes and unifies the theoretical understanding of out-of-distribution detection. Our analytical framework motivates a novel OOD detection method for neural networks, *GEM*, which demonstrates both theoretical and empirical superiority. We formally provide provable guarantees and comprehensive analysis of our method, underpinning how various properties of data distribution affect the performance of OOD detection. We hope our work can motivate future research on the theoretical understandings of OOD detection.

Acknowledgments

The authors would like to thank Yifei Ming for the helpful discussion. This work is supported by the Wisconsin Alumni Research Foundation (WARF), Facebook Research Award, and funding from Google Research.

Bibliography

Bendale, A.; and Boult, T. 2015. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1893–1902.

Bevandić, P.; Krešo, I.; Oršić, M.; and Šegvić, S. 2018. Discriminative out-of-distribution detection for semantic segmentation. *arXiv preprint arXiv:1808.07703*.

Chen, J.; Li, Y.; Wu, X.; Liang, Y.; and Jha, S. 2021. ATOM: Robustifying Out-of-distribution Detection Using Outlier Mining. *In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3606–3613.

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*.

Du, X.; Wang, Z.; Cai, M.; and Li, Y. 2022. VOS: Learning What You Don't Know by Virtual Outlier Synthesis. *Proceedings of the International Conference on Learning Representations*.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*.

Geifman, Y.; and El-Yaniv, R. 2019. Selectivenet: A deep neural network with an integrated reject option. *arXiv preprint arXiv:1901.09192*.

Hein, M.; Andriushchenko, M.; and Bitterwolf, J. 2019. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 41–50.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *Proceedings of International Conference on Learning Representations*.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*.

Huang, R.; Geng, A.; and Li, Y. 2021. On the importance of gradients for detecting distributional shifts in the wild. *Neural Information Processing Systems*.

Huang, R.; and Li, Y. 2021. MOS: Towards Scaling Out-of-distribution Detection for Large Semantic Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8710–8719.

Jaynes, E. T. 1986. *Bayesian methods: General background*. Citeseer.

Jeong, T.; and Kim, H. 2020. OOD-MAML: Meta-learning for few-shot out-of-distribution detection and classification. *Advances in Neural Information Processing Systems*, 33.

Kendall, A.; and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, 5574–5584.

Krizhevsky, A.; Hinton, G.; et al. 2009. *Learning multiple layers of features from tiny images*. Citeseer.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *NeurIPS*.

Lee, K.; Lee, H.; Lee, K.; and Shin, J. 2018a. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. *Proceedings of the International Conference on Learning Representations*.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018b. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 7167–7177.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018*.

Liu, W.; Wang, X.; Owens, J.; and Li, Y. 2020. Energy-based Out-of-distribution Detection. *Advances in Neural Information Processing Systems*.

Mahalanobis, P. C. 1936. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2: 49–55.

Malinin, A.; and Gales, M. 2018a. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, 7047–7058.

Malinin, A.; and Gales, M. 2018b. Predictive uncertainty estimation via prior networks. In *NeurIPS*.

Malinin, A.; and Gales, M. 2019. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *NeurIPS*.

Ming, Y.; Yin, H.; and Li, Y. 2022. On the impact of spurious correlation for out-of-distribution detection. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Mohseni, S.; Pitale, M.; Yadawa, J.; and Wang, Z. 2020. Self-Supervised Learning for Generalizable Out-of-Distribution Detection. In *AAAI*, 5216–5223.

Morteza, P.; and Li, Y. 2021. Provable Guarantees for Understanding Out-of-distribution Detection. arXiv:2112.00787.

Nalisnick, E.; Matsukawa, A.; Teh, Y. W.; Gorur, D.; and Lakshminarayanan, B. 2019. Do deep generative models know what they don't know? *International Conference on Learning Representations*.

Nandy, J.; Hsu, W.; and Lee, M. L. 2020. Towards maximizing the representation gap between in-domain & out-of-distribution examples. In *NeurIPS*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 427–436.

Ren, J.; Liu, P. J.; Fertig, E.; Snoek, J.; Poplin, R.; Depristo, M.; Dillon, J.; and Lakshminarayanan, B. 2019. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 14680–14691.

Serrà, J.; Álvarez, D.; Gómez, V.; Slizovskaia, O.; Núñez, J. F.; and Luque, J. 2020. Input Complexity and Out-of-distribution Detection with Likelihood-based Generative Models. In *International Conference on Learning Representations*.

Sricharan, K.; and Srivastava, A. 2018. Building robust classifiers through generation of confident out of distribution examples. *arXiv preprint arXiv:1812.00239*.

Sun, Y.; Guo, C.; and Li, Y. 2021. ReAct: Out-of-distribution Detection With Rectified Activations. In *Advances in Neural Information Processing Systems*.

Van Amersfoort, J.; Smith, L.; Teh, Y. W.; and Gal, Y. 2020. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, 9690–9700. PMLR.

Vernekar, S.; Gaurav, A.; Abdelzad, V.; Denouden, T.; Salay, R.; and Czarnecki, K. 2019. Out-of-distribution detection in classifiers via generation. *arXiv preprint arXiv:1910.04241*.

Wang, H.; Liu, W.; Bocchieri, A.; and Li, Y. 2021. Can multi-label classification networks know what they don't know? In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Xu, P.; Ehinger, K. A.; Zhang, Y.; Finkelstein, A.; Kulkarni, S. R.; and Xiao, J. 2015. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*.

Yang, J.; Zhou, K.; Li, Y.; and Liu, Z. 2021. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6): 1452–1464.