Deep Metric Learning-Based Out-of-Distribution Detection with Synthetic Outlier Exposure

Assefa Seyoum Wahd

Abstract

In this paper, we present a novel approach that combines deep metric learning and synthetic data generation using diffusion models for out-of-distribution (OOD) detection. One popular approach for OOD detection is outlier exposure, where models are trained using a mixture of in-distribution (ID) samples and "seen" OOD samples. For the OOD samples, the model is trained to minimize the KL divergence between the output probability and the uniform distribution while correctly classifying the in-distribution (ID) data. In this paper, we propose a label-mixup approach to generate synthetic OOD data using Denoising Diffusion Probabilistic Models (DDPMs). Additionally, we explore recent advancements in metric learning to train our models.

In the experiments, we found that metric learning-based loss functions perform better than the softmax. Furthermore, the baseline models (including softmax, and metric learning) show a significant improvement when trained with the generated OOD data. Our approach outperforms strong baselines in conventional OOD detection metrics.

1. Introduction

Out-of-distribution (OOD) detection plays a critical role in the development of robust machine learning models. While accurate classification of known classes is important, the ability to identify samples that deviate from the training distribution is equally crucial. This paper presents a novel approach that combines deep metric learning and synthetic data generation using diffusion models to improve OOD detection in classification models.

One popular approach for OOD detection is outlier exposure [11], which involves training models using seen samples from out-of-distribution data.The model is trained to output a low confidence on the training OOD data while correctly classifying the in-distribution (ID) data. Outlier exposure methods differ in how the OOD data is obtained (real-world data vs. generated data).

[11] is the first successful work to train a classifier using labeled ID data and a large set of unlabeled OOD data. The method hypothesizes that training on a large and diverse OOD data can help deep neural networks (DNNs) generalize better to unseen OOD examples at test time. For example, they train a CIFAR-10 vs. others OOD detector by exposing the model with 80 Million Tiny Images dataset. Outlier exposure outperforms several state-of-the-art OOD detection methods on several benchmark datasets.

However, using a large unlabeled dataset as OOD training data introduces an unwanted problem. Ideally, the training OOD data is considered to have no semantic similarity with the ID data. However, in practice, it is evident that OOD datasets obtained from the wild may contain mixed ID and OOD samples, thus introducing difficulty for outlier exposure methods that use large unlabeled OOD data for training. If the unlabeled dataset contains samples with overlapping semantics with the ID dataset, the network may just focus on minor statistical differences in the images and not the semantic meaning of the images. This may not be desirable as the model can easily overfit to the training ID and OOD datasets. To address this issue, [36] trains an OOD detector by removing any overlapping classes from the OOD dataset by deep clustering [1]. [15] models the mixed dataset as Huber contamination model [14], meaning it is considered to be partially coming from ID and OOD distributions. The model is trained to predict the mixing ratio. Another challenge of utilzing real-world OOD data for training is that that data may not cover the full range of OOD examples that the model may encounter in the real world. Thus, training on OOD data may lead to overfitting on the specific OOD examples used for training. Furthermore, collecting real-world OOD data can be costly, for example, in case of rare events.

Other works have tried to solve the issue of overlapping training ID and OOD semantics and overfitting by generating synthetic OOD data that satisfy certain requirements [18], [24]. The main idea is to generate samples that lie in the low density areas of the training data distribution; i.e., the generated data should be neither too close nor too far from the training data distribution. OOD data that is too close to the training distribution can limit the classifier's closed-set classification accuracy, and data that is too far from the training distribution expands the classifier's decision boundary, possibly classifying OOD datasets as ID.

In this paper, we propose a synthetic OOD data generation approach using denoising diffusion probabilistic Models (DDPMs) [13]. Although this is not the first time diffusion models have been used for out-of-distribution detection, our method differs from previous works in several ways as will be explained next. [24] generates synthetic OOD data by early stopping (i.e., before the model converges) a diffusion model during training. The authors argue that the Fréchet Inception Distance (FID) [12], which measures the quality of the generated image, has a direct correlation to the number of training steps and therefore early stopping ensures that the generated image doesn't fully converge to the training data distribution. The generated data is used to train a binary classifier and the nearest neighbor distance [27] is used as the OOD score. [6] proposed a reconstruction-based novelty detection by first adding a range of noise levels to a given input and then reconstructing it using a pre-trained DDPM. The reconstruction errors between the original and the reconstructed images are computed at several timesteps, and the average reconstruction error is used to classify an input as ID or OOD. [23] used the image inpainting power of diffusion models as the OOD score function. Specifically, a test image is first corrupted by masking a large portion and a diffusion model is used to reconstruct the corrupted image. In-distribution samples have a small reconstruction error (because DDPMs have the ability to inpaint) while the OOD samples typically have a large reconstruction error. We propose a synthetic OOD generation approach by interpolating the one-hot encoding of the target classes in a conditional diffusion model. We refer to this approach as a label mixup. Suppose that we want to a cat vs. dog classifier. First, we train a class-conditional DDPM by giving the one-hot labels (i.e, $[1, 0]$, and $[0, 1]$) and a noise image to the DDPM. To generate a synthetic OOD data, we mix the labels by interpolating the one-hot encodings, i.e., $[1, 1]$. Since, the generative models has only seen $[1, 0]$ or $[0, 1]$ during training, $[1, 1]$ input generates data that lies between the two classes. See Figure 1. It is important to note the previous methods that use DDPM treat OOD detection as a binary classification hence do not train a multi-class classifier which is an important distinction to our work.

It is a standard practice to train multi-class classification models with the softmax loss function. In this work, we take inspiration from the success of contrastive learning methods in OOD detection to investigate alternative loss functions to train our OOD detection models. Contrastive learning [8], [29] trains a similarity function (e.g., cosine similarity [8], [29]) to maximize the similarity between different ("weak") augmentations of a given sample, and minimize the similarity with other samples (i.e., instance discrimination [34]). Specifically, the goal in contrastive learning is to train an encoder neural network such that different random augmentations of the same image are close in the embedding space but far from the embeddings of another image. CSI [29] found that, in addition to pushing away different samples, pushing "strong" augmentations of a sample (e.g., rotation) away from the original sample improves OOD detection as strong augmentation can shift the distribution of an input. To train a model using a contrastive loss function, negative sample mining that chooses for the most useful samples is crucial. In CSI, positive samples of an input are obtained with weak augmentations (e.g., cropping) while negative samples include strong augmentations of the same image as well as other images from the training dataset. CADet [8] utilized the maximum mean discrepancy (MMD) two-sample test [7] as a score function for models trained with contrastive loss functions. Angle-based metric learning methods ([22],[32],[41],[2]), on the other hand, propose a similarity learning mechanism without negative sample mining. Metric (distance) learning techniques are commonly employed to increase inter-class variation and reduce intra-class variation in the feature space of deep neural networks, especially in few-shot settings [28],[20],[40],[37] and deep face recognition [22],[32],[41],[2]. In this work, we regard state-of-the-art metric learning loss functions, such as SphereFace [22], CosFace [32], AdaCos [41], and ArcFace [2] as OOD score functions.

To evaluate the effectiveness of our method, we compare it with well-known approaches in the field. Our results show that our approach outperform baseline methods in conventional OOD detection metrics (AUROC and AUPR). By combining deep metric learning and synthetic data generation, our proposed method offers a promising solution for improving OOD detection.

In summary the contributions of this paper are,

- We introduce a synthetic out-of-distribution data generation using denoising diffusion models.
- We adapt popular loss functions in deep metric learning for out-of-distribution detection.
- We show that models trained with the proposed outlier exposure outperform the regular softmax and metric learning loss function.
- To the best of our knowledge this is the first time diffusion models have been used to generate synthetic OOD data by label mixup and used to train a multi-class classifier.

2. Related Work

Maximum-Softmax Probability (MSP) [10]. This baseline method uses the highest output probability as the score function. The intuition is that a classifier should be more confident about in-distribution inputs than OOD inputs. The MSP score function is defined as follows:

$$s_\theta(x) = \max_{c \in C} p_\theta(y = c|x). \tag{1}$$

Energy-Based OOD Detection (EBO) [21]. In EBO, an energy score is derived as the 'logsumexp' of the output predictions scaled by a temperature T:

$$s_\theta(x) = T \log \sum_{i}^{C} \exp\left(f_\theta(x; i)/T\right) \tag{2}$$

where $f_\theta(x; i)$ is the logit value corresponding the $i - th$ class of the classifier f_θ.

Mahalanobis Distance [19]. The Mahalanobis distance measures of how far a point is from the mean of a distribution. Firstly, class-conditional Gaussian distributions are formed from the features of the penultimate layer, with $\mu_c = \frac{1}{N_c} \sum_{i:y_i=c}^{N_c} f_\theta(x_i)$, for $c = 1, \ldots, C$, a covariance matrix, $\Sigma = \frac{1}{N_c} \sum_{c=1}^{C} \sum_{i:y_i=c} (f_\theta(x_i) - \mu_c)(f_\theta(x_i) - \mu_c)^T$. Then the OOD score function, $s_\theta(x)$, is defined as the negative of the minimum distance from each conditional Gaussian distribution:

$$s_\theta(x) = -\min_{c \in C} (f_\theta(x_i) - \mu_c)\Sigma^{-1}(f_\theta(x_i) - \mu_c)^T \tag{3}$$

Outlier Exposure. [11] trains a classifier using labeled ID data and a large set of unlabeled OOD data. The models have better calibration and OOD detection ability. The problem of overlapping semantics between the training ID data and OOD data has been studied in [36]. The method uses deep clustering [1] to filter out the semantically overlapping samples from the unlabeled OOD data. [15] models the training OOD dataset as Huber contamination model [14], meaning it is considered to be partially coming from ID and OOD distributions. The model is trained to predict the mixing ratio. [4] studies the effectiveness of pre-trained transformer models for out-of-distribution detection. Their findings demonstrate that large scale pre-trained transformer models fine-tuned on the ID data have excellent discriminative ability, but not a well-separated boundary for OOD detection. To improve the OOD detection capability of such models the authors fine-tune the model on seen OOD samples. The key takeway is that setting aside a training OOD data is important to detect OOD samples at test time.

Synthetic Data Outlier Exposure. Synthetic outlier exposure uses generated data as the seen OOD data. [18] jointly trains a classifier and a generative adversarial network (GAN) [5] where the classifier is trained to correctly classify the ID dataset but output a low confidence for the generated dataset. The generator is supervised not only by the discriminator but also by the classifier. This ensures that the generated dataset is neither too far nor too close to the training distribution. This is generally the essence of a synthetic OOD data; i.e., the generated data should approach the training distribution but not too close. [31] generates two types of OOD data using a conditional variational autoencoders (VAEs) [16]: samples that are close to the in-distribution but outside the in-distribution manifold and samples are in the in-distribution manifold but near the in-distribution boundary. The method trains a $K + 1$ classifier, where K is the number of classes and the $K + 1 - th$ class represents the OOD class. Virtual outlier synthesis (VOS) [3] proposes to dynamically generate virtual outliers from low-likelihood region of the Gaussian distribution formed from the empirical means and standard deviations of the features in the penultimate layer of the classification model. Another closely related work [33] generates synthetic data by linearly interpolating the one-hot encodings of target classes in the training data to form pseudo class embeddings and generate an image by feeding the resulting embedding to the decoder network of a variational autoencoders (VAEs) [16]. This is similar to our data generation approach except we use diffusion models as opposed to VAEs.

Diffusion models for OOD detection. Recently, diffusion models have been used for unsupervised anomaly detection [24], [6], [23]. [24] generates synthetic OOD data by early stopping, [6] used the reconstruction error of a noised image as the score function, and [23] intentionally corrupts the input image by cutting out a large portion of the image and reconstruct it using a pre-trained DDPM. The L2 distance between the original image and the reconstructed image is used as the OOD detection score function.

3. The Proposed Method

Let $f_\theta(x)$ be a multi-class classification model where it takes an input $x \in \mathbb{R}^d$, and predicts a vector of probabilities $p_\theta(y|x) \in [0, 1]^C$, where d is the number of features and C is the number of classes. Deep neural networks trained on a dataset $X = \{x_1, \ldots, x_n\} \sim p_{data}(x)$ tend to make an overconfident prediction when exposed to previously unseen distribution, $p_{OOD}(x)$. Out-of-distribution detection aims to detect whether an input x comes from $p_{data}(\cdot)$ or $p_{OOD}(\cdot)$.

Let $s_\theta(x) \in \mathbb{R}$ be a score function that assigns a higher value to in-distribution (ID) inputs and a lower value to out-of-distribution (OOD) inputs. The score function is used to measure how likely an input x is to come from the training data distribution. If the score $s_\theta(x)$ is low, then the input x is likely to be OOD data.

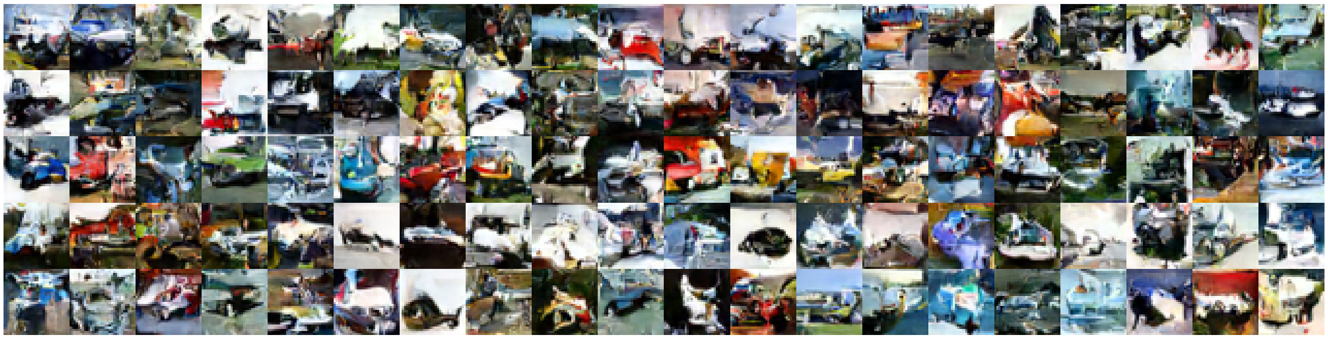

Figure 1: Synthetic OOD data generated using label mixup between CIFAR-10 “airplane” and “automobile” classes. The generated data have a significant diversity and meaningful mixup semantics. For example, a mixup between an airplane class and a automobile class results in an object with features from airplane and automobile.

3.1. Out-of-Distribution Data Generation using Diffusion Models

To generate OOD data, we interpolate between the one-hot encoding vectors of any two different classes, which we refer to as label mixup. By doing so, we create new pseudo class embeddings that represent OOD data. These embeddings represent images that contain features from both classes and can be used to generate high-quality synthetic data. The resulting vector is mapped to the pixel space using a conditional DDPM pre-trained on the in-distribution data.

To explain why this works, we present an analogy between our proposed label mixup and mixup training [39]. Mixup training is a regularization (data augmentation) technique that trains a neural network on the convex combinations of pairs of examples and their labels. By doing so, mixup regularizes the neural network to favor simple linear behavior between training examples. The decision boundary in a model trained with mixup regularization smoothly decays from one class to another, thus predicting low confidence for data that lies in between. It has been shown that mixup training generalizes to out-of-distribution and adversarial examples. Similarly, label mixup can be considered as a form of mixup but in the label space instead of the image space.

To this end, we select any two different classes and add their one-hot encodings element-wise, and input the resulting label (in addition to a noise input sampled from a uniform distribution) to a pre-trained DDPM. To train the DDPM we use the pipieline from Hugging Face diffusers library[1]. Interested readers can refer to [13] for more details about diffusion models. See Figure 1 for a snapshot of the generated data. The training scheme for the OOD detector will follow next.

3.2. Deep Metric Learning

The objective is to learn an encoder neural network f_θ such that samples of the same class are close in the embedding space while samples of different classes are far in the embedding space. In other words, we require small intra-class and a large inter-class variance. Suppose we have a neural network $f_\theta(x_i) \rightarrow z_i \in \mathbb{R}^d$, where x_i is an input image and z_i represents the features in the penultimate layer. Normally a fully connected layer with weights $W \in \mathbb{R}^{d \times C}$ and biases $b \in \mathbb{R}^C$ is used to project z_i into the logit space and the softmax (cross-entropy) loss is derived as follows:

$$\mathcal{L}_{\text{softmax}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{W_{y_i}^T z_i + b_{y_i}}}{\sum_{j=1}^{C} e^{W_{y_j}^T z_i + b_{y_j}}}, \tag{4}$$

where W_{y_i}, b_{y_i} are the weights and the bias associated with the class y_i, and C is the number of classes. If we fix $b_{y_i} = 0$ and normalize the weights s.t. $|W_{y_i}| = 1$, we can rewrite $W_{y_j}^T z_i + b_{y_j}$ as $|z_i|\cos(\theta_{j,i})$, where $\theta_{j,i}$ is the angle between the feature vector z_i and the class weights W_{y_j}. $|z_i|\cos(\theta_{j,i})$ is the projection of z_i onto the class weights W_{y_j}. x_i is classified as class y_i if $\cos(\theta_{i,i}) > \cos(\theta_{j,i}), \forall j \in 1,\dots,C$. SphereFace [22] introduces a margin m s.t. x_i is classified as class y_i if $\cos(m\theta_{i,i}) > \cos(\theta_{j,i}), \forall j \in 1,\dots,C$. This encourages a larger inter-class distance as it moves the decision boundary from a bisector between W_i and W_j to an angular margin m. The softmax loss with the proposed angular margin becomes:

$$\mathcal{L}_{\text{angular}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{|z_i|\cos(m\theta_{i,i})}}{e^{|z_i|\cos(m\theta_{i,i})} + \sum\limits_{j \neq i} e^{|z_i|\cos(\theta_{j,i})}}, \tag{5}$$

In this work, we regard state-of-the-art metric learning loss functions, such as SphereFace [22], CosFace [32], ArcFace [2], and AdaCos [41] as OOD score functions. We will briefly describe each loss function. A comprehensive anal-

[1] https://huggingface.co/docs/diffusers/index

ysis of each loss function is beyond the scope of this paper, and readers are encouraged to refer to the respective papers for detailed derivations. For our purposes, we treat the loss functions the same; we use the maximum cosine similarity between the features z_i and the weight vectors $W_i, \forall i \in 1, \ldots, C$, as the OOD score function. In addition to normalizing the weight vectors $W_{y_i}, \forall i = 1, \ldots, C$, CosFace [32] normalizes the features vectors z_i s.t. $|z_i| = 1$. The CosFace loss function is defined as follows:

$$\mathcal{L}_{\text{CosFace}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s(\cos(\theta_{i,i})-m)}}{e^{s(\cos(\theta_{i,i})-m)} + \sum_{j \neq i} e^{s\cos(\theta_{j,i})}}, \tag{6}$$

where s is a scaling factor and m is the margin. ArcFace [2], like CosFace, normalizes both the weights and the features. However, the angular margin is defined in the angle space as given in the following equation:

$$\mathcal{L}_{\text{ArcFace}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s\cos(\theta_{i,i}+m)}}{e^{s\cos(\theta_{i,i}+m)} + \sum_{j \neq i} e^{s\cos(\theta_{j,i})}}, \tag{7}$$

Adacos [41] proposed a fixed scaling parameter s defined by the following equation:

$$s \approx \sqrt{2}\log(C-1), \tag{8}$$

where C is the number of classes. The loss function remains the sames as Eqn. (7).

3.3. OOD Detector Training

In addition to the synthetic data generation approach, our contribution is using the deep metric learning loss functions for training OOD detectors. We train two types of models: with and without synthetic outlier exposure. Both types of models are trained with the vanilla softmax loss function (Eqn. (4)) and the metric learning-based loss functions (e.g., Eqn. (5)).

Metric learning-based OOD detection have been studied before in [26], and [30]. [30] uses the scaled cosine similarity as the score function. Specifically, the weights and features are normalized to be a unit vector and their dot product scaled by a learnable parameter s is used as the score function, i.e., $\cos(\theta_{i,j}) = W_{y_j}^T z_i / |W_{y_j}||z_i|$, and $\mathcal{L} = -\frac{1}{N}\sum_{i}^{N} \log \frac{e^{s\cos(\theta_{i,i})}}{\sum_{j=1}^{C} s\cos(\theta_{i,j})}$, where z_i represents the features of input x_i and W_{y_j} are the weights of the class $j - th$ class. In this study, we aim to explore more metric learning loss functions including the scaled cosine loss function and compare their OOD detection performance before and after the proposed outlier exposure.

3.4. Detecting OOD samples

A test sample x is predicted as OOD if the maximum cosine similarity between the normalized features and weights is less than a threshold τ, otherwise it is predicted as ID. The threshold is computed from a validation set at 95% true positive rate (TPR). If x is predicted as ID, we predict its class as the index with the maximum cosine similarity scaled by s and m. For example, when using the SphereFace loss function (Eqn. (5)), the index with the highest $\cos(m\theta_{i,i})$ becomes the predicted class. For models trained with the softmax loss function, the maximum probability used to decide if x is ID or OOD.

4. Experiments

We use the ResNet-50 [9] architecture to train all models. We use CIFAR-10 datasets as the in-distribution dataset for all experiments and we use CIFAR-100, Tiny ImageNet [17], SVHN [25], iSUN [35], LSUN [38] as out-of-distribution datasets.

4.1. Experimental Results

We compare the AUROC, AUPR-In (when the in-distribution data is the positive class) and AUPR-Out (when the OOD data is the positive class) results of the baseline models and our models in Table 1. We use the model trained with the softmax loss function and the MSP [10] score function as a baseline and compare it against our models trained with the metric learning-based loss functions. In Table 1 (top), we show the models before using synthetic outlier exposure and the performance after synthetic outlier exposure in Table 1 (bottom). The models with the synthetic data show a significant performance gain in both the vanilla softmax and the metric learning loss functions. Notably, when the baseline models including the softmax loss function and the metric loss functions struggle with certain datasets such as Gaussian noise, uniform noise, and Tin (R), the models with the outlier exposure produce consistent results across all datasets. Furthermore, the scaled cosine (which is also metric learning-based loss function) outperforms the softmax based training.

In summary, the results show that the proposed data generation approach generalizes across several training loss functions, usually with a significant improvement.

Closed-Set Accuracy We compare the in-distribution classification accuracy of the baseline models (i.e., trained on ID data only) and their performance after synthetic outlier exposure. Table 2 illustrates that the OOD detectors' closed-set accuracy is comparable to that of the baseline classifiers. Our models have the ability to detect out-of-distribution samples with a small drop in closed-set classification accuracy.

	Method	Softmax	Scaled Cosine	AdaCos (ours)	ArcFace (ours)	CosFace (ours)	SphereFace (ours)
Without Outlier Exposure	CIFAR-100	**0.913/0.896/0.892**	0.869/0.871/0.868	0.885/0.868/0.884	0.878/0.858/0.878	0.844/0.806/0.856	0.874/0.851/0.879
	LSUN (C)	0.937/0.917/0.928	**0.977/0.977/0.978**	0.969/0.965/0.970	0.966/0.963/0.967	0.965/0.960/0.968	0.974/0.973/0.975
	LSUN (R)	**0.962/0.967**/0.949	0.961/0.963/**0.961**	0.942/0.939/0.942	0.937/0.931/0.939	0.943/0.945/0.942	0.941/0.935/0.946
	Tin (C)	0.960/0.964/0.945	**0.965/0.968/0.963**	0.928/0.928/0.927	0.912/0.898/0.920	0.931/0.921/0.934	0.934/0.929/0.938
	Tin (R)	**0.950/0.952**/0.932	0.945/0.950/0.942	0.880/0.872/0.883	0.858/0.833/0.873	0.895/0.885/0.896	0.886/0.874/0.896
	iSun	**0.963/0.971**/0.944	0.956/0.961/**0.952**	0.925/0.925/0.920	0.919/0.915/0.917	0.940/0.946/0.932	0.934/0.935/0.932
	SVHN	0.955/0.925/0.973	0.966/0.930/0.986	**0.978/0.951/0.991**	0.968/0.943/0.986	0.951/0.853/0.982	0.968/0.947/0.985
	Gaussian Noise	0.837/0.878/0.700	**1.000/1.000/1.000**	0.999/0.999/0.998	0.999/0.999/0.997	**1.000/1.000/1.000**	0.999/0.999/0.997
	Uniform Noise	0.923/0.951/0.847	**1.000/1.000/1.000**	1.000/1.000/1.000	0.999/0.999/0.996	**1.000/1.000/1.000**	**1.000/1.000**/0.998
With Outlier Exposure	CIFAR-100	0.919/0.919/0.902	**0.934/0.935/0.929**	0.894/0.885/0.883	0.902/0.895/0.891	0.902/0.906/0.891	0.888/0.885/0.879
	LSUN (C)	0.975/**0.980**/0.970	**0.983/0.985/0.982**	0.972/0.975/0.971	**0.976**/0.978/**0.974**	0.974/0.976/0.973	0.969/0.972/0.968
	LSUN (R)	0.984/0.986/0.983	**0.990/0.991/0.990**	0.978/0.981/0.975	0.981/0.984/0.980	0.972/0.975/0.970	0.973/0.975/0.972
	Tin (C)	0.974/0.979/0.968	**0.985/0.987/0.983**	0.955/0.960/0.948	0.967/0.973/0.962	0.961/0.966/0.957	0.944/0.945/0.942
	Tin (R)	0.973/0.977/0.968	**0.984/0.985/0.983**	0.953/0.956/0.948	0.965/0.969/0.960	0.952/0.958/0.947	0.935/0.933/0.935
	iSun	0.980/0.985/0.975	**0.988/0.990/0.987**	0.968/0.976/0.960	0.974/0.980/0.967	0.965/0.972/0.958	0.962/0.968/0.957
	SVHN	0.962/0.943/0.978	**0.985/0.974/0.993**	0.964/0.936/0.983	0.979/0.957/0.957	0.969/0.944/0.986	0.976/0.954/0.989
	Gaussian Noise	0.976/0.985/0.951	**0.999/0.999/0.999**	**0.999/0.999/0.999**	**0.999/0.999**/0.996	0.999/0.999/0.995	**0.999/0.999**/0.997
	Uniform Noise	0.985/0.991/0.968	**0.999/1.000/0.999**	**0.999**/0.999/**0.999**	**0.999/0.999**/0.996	0.999/0.999/0.999	**0.999/0.999**/0.997

Table 1: Out-of-distribution detection evaluation results before and after outlier exposure. The numbers separated by / indicate AUROC/AUPR-In/AUPR-Out. **Boldface** indicates the best approach and underline (_) indicates the second best. The amount of increase/decrease from the baseline (without outlier exposure) to the models trained with outlier exposure is indicated in the last section (difference) of this table.

	Softmax	Scaled Cosine [30]	AdaCos [41]	ArcFace [2]	CosFace [32]	SphereFace [22]
Standard	**96.98**	92.28	**96.39**	**96.41**	**96.08**	**96.21**
Outlier Exposure (ours)	96.54	**96.78**	95.80	95.81	95.61	95.70

Table 2: Our proposed outlier exposure has minimal impact on the in-distribution classification accuracy as it remains largely unchanged. We highlight in **bold** the model with the higher in-distribution accuracy for each loss function.

5. Conclusion

In conclusion, this paper introduced a novel method for out-of-distribution (OOD) detection in classification models by combining deep metric learning and synthetic data generation using diffusion models. The approach employs outlier exposure, a popular technique for OOD detection, where models are trained using known OOD samples. During training, the model low confidence for the training OOD data, while accurately classifying the in-distribution (ID) data. To generate synthetic OOD data, we proposed a label-mixup approach using Denoising Diffusion Probabilistic Models (DDPMs), and we utilize recent advancements in metric learning to train our models.

The experimental results demonstrate that our method, employing outlier exposure with metric learning, outperforms softmax training in most settings. Moreover, all loss functions including the vanilla softmax and metric learning-based loss functions show a significant improvement after the proposed outlier exposure.

The experimental results demonstrate that our method, employing outlier exposure shows a significant improvement across a range of loss functions and datasets compared to the baseline models that do not have access to a training OOD data.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[2] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 741–757. Springer, 2020.

[3] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*, 2022.

[4] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:7068–7081, 2021.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances*

in neural information processing systems, pages 2672–2680, 2014.

[6] Mark S Graham, Walter HL Pinaya, Petru-Daniel Tudosiu, Parashkev Nachev, Sebastien Ourselin, and Jorge Cardoso. Denoising diffusion models for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2947–2956, 2023.

[7] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[8] Charles Guille-Escuret, Pau Rodriguez, David Vazquez, Ioannis Mitliagkas, and Joao Monteiro. Cadet: Fully self-supervised anomaly detection with contrastive learning. *arXiv preprint arXiv:2210.01742*, 2022.

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

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

[11] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, pages 6840–6851, 2020.

[14] Peter J Huber. Robust estimation of a location parameter. *Breakthroughs in statistics: Methodology and distribution*, pages 492–518, 1992.

[15] Julian Katz-Samuels, Julia B Nakhleh, Robert Nowak, and Yixuan Li. Training ood detectors in their natural habitats. In *International Conference on Machine Learning*, pages 10848–10865. PMLR, 2022.

[16] Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2013.

[17] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[18] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

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

[20] Xiaoxu Li, Jijie Wu, Zhuo Sun, Zhanyu Ma, Jie Cao, and Jing-Hao Xue. Bsnet: Bi-similarity network for few-shot fine-grained image classification. *IEEE Transactions on Image Processing*, 30:1318–1331, 2020.

[21] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020.

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

[23] Zhenzhen Liu, Jin Peng Zhou, Yufan Wang, and Kilian Q Weinberger. Unsupervised out-of-distribution detection with diffusion inpainting. *arXiv preprint arXiv:2302.10326*, 2023.

[24] Hossein Mirzaei, Mohammadreza Salehi, Sajjad Shahabi, Efstratios Gavves, Cees GM Snoek, Mohammad Sabokrou, and Mohammad Hossein Rohban. Fake it until you make it: Towards accurate near-distribution novelty detection. In *The Eleventh International Conference on Learning Representations*, 2022.

[25] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[26] Deepak Ravikumar, Sangamesh Kodge, Isha Garg, and Kaushik Roy. Intra-class mixup for out-of-distribution detection. *IEEE Access*, 11:25968–25981, 2023.

[27] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR, 2022.

[28] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[29] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 33:11839–11852, 2020.

[30] Engkarat Techapanurak, Masanori Suganuma, and Takayuki Okatani. Hyperparameter-free out-of-distribution detection using cosine similarity. In *Proceedings of the Asian conference on computer vision*, 2020.

[31] Sachin Vernekar, Ashish Gaurav, Vahdat Abdelzad, Taylor Denouden, Rick Salay, and Krzysztof Czarnecki. Out-of-distribution detection in classifiers via generation. *arXiv preprint arXiv:1910.04241*, 2019.

[32] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[33] Mengyu Wang, Yijia Shao, Haowei Lin, Wenpeng Hu, and Bing Liu. Cmg: A class-mixed generation approach to out-of-distribution detection. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML*

PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part IV, pages 502–518, 2023.

[34] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

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

[36] Jingkang Yang, Haoqi Wang, Litong Feng, Xiaopeng Yan, Huabin Zheng, Wayne Zhang, and Ziwei Liu. Semantically coherent out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8301–8309, 2021.

[37] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. Dpgn: Distribution propagation graph network for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13390–13399, 2020.

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

[39] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[40] Hongguang Zhang, Piotr Koniusz, Songlei Jian, Hongdong Li, and Philip HS Torr. Rethinking class relations: Absolute-relative supervised and unsupervised few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9432–9441, 2021.

[41] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019.