OpenGAN: Open-Set Recognition via Open Data Generation

Shu Kong*, Deva Ramanan*,†
*Carnegie Mellon University †Argo AI
{shuk, deva}@andrew.cmu.edu

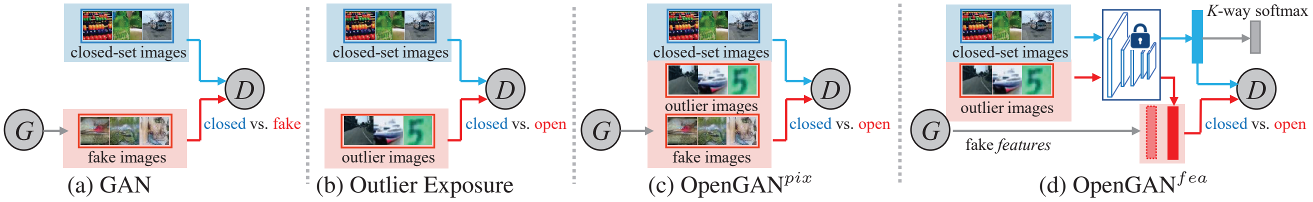

Figure 1: We explore open-set recognition, which requires the ability to discriminate open-set test examples outside K classes of interest. **(a)** Past work has suggested that GAN discriminators can serve as open-set likelihood functions, but this does not work well due to instable training of GANs [47, 44, 39, 56, 30]. **(b)** Outlier Exposure [25] exploits some outlier data to learn a binary discriminator D for open-set discrimination. Because outliers observed during training will not exhaustively span the open-world, the discriminator D tends to generalize poorly to diverse open data [48]. **(c)** We introduce OpenGAN, which augments training outliers with *fake* open data synthesized by a generator G trained to fool the discriminator D. Importantly, we find that a small number of outliers stabilizes training by enabling effective model selection of the discriminator D. **(d)** Because we are concerned with accurate discrimination rather than realistic pixel generation, we find it more efficient to generate (and discriminate) *features* from the off-the-shelf K-way classification network. This allows OpenGAN to be implemented via a lightweight discriminator head built on top of an existing K-way network, enabling closed-world systems to be readily modified for open-set recognition.

Abstract

Real-world machine learning systems need to analyze novel testing data that differs from the training data. In K-way classification, this is crisply formulated as open-set recognition, core to which is the ability to discriminate open-set data outside the K closed-set classes. Two conceptually elegant ideas for open-set discrimination are: 1) discriminatively learning an open-vs-closed binary discriminator by exploiting some outlier data as the open-set, and 2) unsupervised learning the closed-set data distribution with a GAN and using its discriminator as the open-set likelihood function. However, the former generalizes poorly to diverse open test data due to overfitting to the training outliers, which unlikely exhaustively span the open-world. The latter does not work well, presumably due to the instable training of GANs. Motivated by the above, we propose OpenGAN, which addresses the limitation of each approach by combining them with several technical insights. First, we show that a carefully selected GAN-discriminator on some real outlier data already achieves the state-of-the-art. Second, we augment the available set of real open training examples with adversarially synthesized "fake" data. Third and most importantly, we build the discriminator over the features computed by the closed-world K-way networks. Extensive experiments show that OpenGAN significantly outperforms prior open-set methods.

1. Introduction

Machine learning systems that operate in the real open-world invariably encounter test-time data that is unlike training examples, such as anomalies or rare objects that were insufficiently or even never observed during training. Fig. 2 illustrates two cases in which a state-of-the-art semantic segmentation network misclassifies a "stroller"/ "street-market" — a rare occurrence in either training or testing — as a "motorcycle"/"building". This failure could be catastrophic for an autonomous vehicle.

Addressing the open-world has been explored through anomaly detection [59, 25] and out-of-distribution detection [29]. In K-way classification, it can be crisply formulated as open-set recognition, which requires discriminating open-set data that belongs to a $(K{+}1)^{th}$ "other" class, outside the K closed-set classes [45]. Typically, open-set discrimination assumes no training examples from the "other" class (i.e., open-training data) [5, 54, 35]. In this setup, an elegant idea is to learn the closed-set data distribution with a GAN and use a GAN-discriminator as the open-set likelihood function (Fig. 1a) [47, 44, 39, 56, 30]. However, it does not work well due to instable training of GANs. Recent work has shown that outlier exposure (Fig. 1b), or the ability to train on *some* outlier data as open-training examples, can work surprisingly well via the training of a simple open-vs-closed binary discriminator [15, 25]. However,

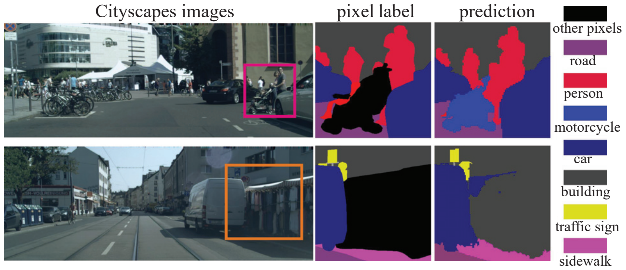

Figure 2: We motivate open-set recognition with safety concerns in autonomous vehicles (AVs). Contemporary benchmarks such as Cityscapes [11] ignore a sizeable "other" pixels for evaluation, which are outside K classes of interest. As a result, most state-of-the-art segmentation approaches [52] also ignore them during training, which then become open-set examples. Perhaps surprisingly, the ignored "other" pixels include vulnerable objects like wheelchairs and strollers (upper row). Here, a semantic segmentation network [52] does not model strollers (upper row) or street-market (lower row), which are outside the K closed-set classes in Cityscapes. The network misclassifies the stroller as a "motorcycle", and the street-market as "building". Such misclassifications can be a critical mistake when fed into AVs because these objects require different plans for obstacle avoidance (e.g., "yield" or "slow-down"). Fig. 4 shows qualitative results by our approach.

such discriminators generalize poorly to diverse open-set data [48] due to overfitting to the available set of training outliers, which are often biased and fail to exhaustively span the open-world. Motivated by above, we introduce **OpenGAN**, a simple approach that dramatically improves open-set classification accuracy by incorporating several key insights. First, we show that using outlier data as a valset to select the "right" GAN-discriminator *does* achieve the state-of-the-art on open-set discrimination. Second, with outlier exposure, we augment the available set of open-training data by adversarially generating *fake* open examples that fool the binary discriminator (Fig. 1c). Third and most importantly, rather than defining discriminators on pixels, we define them on off-the-shelf (OTS) features computed by the closed-world K-way classification network (Fig. 1d). We find such discriminators generalize much better.

Our formulation differs in three ways from other open-set approaches that employ GANs. (1) Our goal is *not* to generate realistic pixel images, but rather to learn a robust open-vs-closed discriminator that naturally serves as an open-set likelihood function. Because of this, our approach might be better characterized as a discriminative adversarial network! (2) We train the discriminator with both *fake* data (synthesized from the generator) and *real* open-training examples (cf. outlier exposure [25]). (3) We train GANs on OTS features rather than RGB pixels. We show that OpenGAN significantly outperforms prior work for open-set recognition across a variety of tasks including image classification and pixel segmentation. Moreover, we demonstrate that our technical insights improve the accuracy of other GAN-based open-set methods: training them on OTS features and selecting their discriminators via validation as the open-set likelihood function.

2. Related Work

Open-Set Recognition. There are multiple lines of work addressing open-set discrimination, such as anomaly detection [9, 29, 59], outlier detection [44, 39], and open-set recognition [45, 19]. The typical setup for these problems assumes that one does not have access to training examples of open-set data. As a result, many approaches propose to first train a closed-world K-way classification network on the closed-set [24, 5] and then exploit the trained network for open-set discrimination [45, 28, 35]. Some others train "ground-up" models for both closed-world K-way classification and open-set discrimination by synthesizing fake open data during training, oftentimes sacrificing the classification accuracy on the closed-set [18, 33, 54, 50]. To recognize open-set examples, they resort to post-hoc functions like density estimation [59, 57], uncertainty modeling [17, 29], and input image reconstruction [39, 20, 14, 50]. We also explore open-set recognition through K-way classification networks, but we show OpenGAN, a simple and direct method of training an open-vs-closed classifier on adverserial data, performs significantly better than prior work.

Open-Set Recognition with GANs. As GANs can learn data distributions [21], conceptually, a GAN-discriminator trained on the closed-set naturally serves as an open-set likelihood function. However, this does not work well [47, 44, 39, 56, 30], presumably due to instable training of GANs. As a result, previous GAN-based methods focus on 1) generating fake open-set data to augment the training set, and 2) relying on the reconstruction error for open-set recognition [53, 47, 44, 1, 13]. With OpenGAN, we show that GAN-discriminator can achieve the state-of-the-art for open-set discrimination *once* we perform model selection on a valset of outlier examples. Therefore, unlike prior approaches, OpenGAN directly uses the discriminator as the open-set likelihood function. Moreover, our final version of OpenGAN generates features rather than pixel images.

Open-Set Recognition with Outlier Exposure. [15, 25, 43] reformulate the problem with the concept of "outlier exposure" which allows methods to access *some* outlier data as open-training examples. In this setting, simply training a binary open-vs-closed classifier works surprisingly well. However, such classifiers easily overfit to the available set of open-training data and generalize poorly, e.g., in a "cross-dataset" setting where open-set testing data differs from open-training data [48]. It appears fundamentally challenging to collect outlier data to curate an exhaustive training set of open-set examples. Our approach, OpenGAN, attempts to address this issue by augmenting the training set with adversarial *fake* open-training examples.

3. OpenGAN for Open-Set Recognition

Generally, solutions to open-set recognition contain two steps: (1) open-set discrimination that classifies testing examples into closed and open sets based on the open-set likelihoods, and (2) K-way classification on closed-set from step (1) [45, 4, 35]. The core problem to open-set recognition is the first step, i.e., open-set discrimination. Typically, open-set discrimination assumes no availability of open-set training data [33, 35]. [15, 25] convincingly demonstrate that outlier exposure, or the ability to train on *some* outlier examples as open-training data, can greatly improve open-set discrimination via the training of a simple open-vs-closed binary classifier (Fig. 1b). Because it is challenging to construct a training set that exhaustively spans the open-world, such a classifier may overfit to the outlier data and not sufficiently generalize [48]. We demonstrate that OpenGAN alleviates this challenge by generating fake open-set training examples using a generator that is adversarially trained to fool the classifier. Importantly, with model selection on a valset, OpenGAN is also effective under the classic setup which assumes no availability of open-training data.

3.1. Methodology

Let x be a data example, which can be an RGB image or its feature representation. We will show that using the latter performs better. Let $\mathcal{D}_{closed}(x)$ be the closed-world distribution over x — that is, closed-set data from the K closed-set classes. Let $\mathcal{D}_{open}(x)$ be the open-set data distribution of examples which do not belong to the closed-set.

Binary Classifier. We train a binary classifier D from both closed- and open-set data:

$$\max_{D} \mathbb{E}_{x\sim\mathcal{D}_{closed}}[\log D(x)] + \lambda_o \cdot \mathbb{E}_{x\sim\mathcal{D}_{open}}[\log(1 - D(x))]$$

where $D(x) = p(y\text{=“closed-set”}|x)$. Intuitively, we tune λ_o to balance the closed- and open-set training examples. This simple method is effective when the open-training examples are sufficiently representative of testing-time open-set data [25], but underperforms when they fail to span the open-world [48].

Synthetic Open Data. One solution to the above is to exploit synthetic data, armed with which we might expect the classifier D to perform better. Assume we have a generator network $G(z)$ that produces synthetic images given (Gaussian normal) random noise inputs $z \sim \mathcal{N}$. We can naively add them to the pool of negative or open-set examples that D should not fire on. But these synthetic images might be too easy for D to categorize as open-set data [32, 10]. A natural solution is to adversarially train the generator G to produce difficult examples that fool the classifier D using a GAN loss:

$$\min_{G} \mathbb{E}_{z\sim\mathcal{N}}\Big[\log(1 - D(G(z)))\Big] \tag{1}$$

Because a *perfectly* trained generator G would generate realistic closed-set images, eventually making the discriminator D inapplicable for open-set discrimination. We find that the following two techniques easily resolve this issue.

OpenGAN trains with both the *real* open&closed-set data and the *fake* open-data into a single (GAN-like) minimax optimization over D and G:

$$\begin{aligned}\max_{D}\min_{G} & \mathbb{E}_{x\sim\mathcal{D}_{closed}}[\log D(x)] \\ & + \lambda_o \cdot \mathbb{E}_{\tilde{x}\sim\mathcal{D}_{open}}[\log(1 - D(\tilde{x}))] \\ & + \lambda_G \cdot \mathbb{E}_{z\sim\mathcal{N}}[\log(1 - D(G(z)))]\end{aligned} \tag{2}$$

where λ_G controls the contribution of generated fake open-data by G. When there are no open training examples (i.e., λ_o=0), the above minimax optimization can still train a discriminator D for open-set classification. In this case, training an OpenGAN is equivalent to training a normal GAN and using its discriminator as the open-set likelihood function. While the literature finds GAN-discriminator to not work well, we show it *does* achieve the state-of-the-art *once* it is selected using a valset (detailed below). To distinguish our contribution on the crucial step of model selection via validation, we call this method OpenGAN-0.

Open Validation. Model selection is challenging for GANs. Typically, one resorts to visual inspection of generated images from different model checkpoints to select the generator G [21]. In our case, we must carefully select the discriminator D. We experimented with many approaches such as using the last model checkpoint or selecting the one with minimum training error, but neither works, because adversarial training will eventually lead to a discriminator D that is incapable of discriminating closed-set data and fake open-set data generated by G (details in the supplemental). We find it crucial to use a validation set of real outlier data to select D, when D achieves the best open-vs-closed classification accuracy on the valset. We find the performance to be quite robust to the val-set of outlier examples, even when they are drawn from a different distribution from those encountered at test-time (Table 3 and 4).

3.2. Further Discussion on Prior GAN Methods

Numerous works have used GANs for open-set discrimination. We compare OpenGAN to this literature.

Discriminator vs. Generator. GANs mostly aim at generating realistic images [2, 7]. As a result, prior work in open-set recognition has focused on using GANs to generate realistic open-training images [18, 27, 33]. These additional images are used to augment the training set for learning an open-set model, which oftentimes is designed for both the closed-world K-way classification and open-set discrimination [18, 27, 33]. In our case, we do not learn a separate open-set model but directly use the already-trained discriminator *as* the open-set likelihood function.

Features vs. Pixels. GANs are typically used to generate realistic pixel images. As a result, many GAN-based open-set methods focus on generating realistic images to augment the closed-set training data [39, 56, 25]. However, generating high-dimensional realistic images is challenging per se [2, 7] and may not be necessary to open-set recognition [39]. As such, we build GANs over OTS feature embeddings learned by the closed-world K-way classification networks, e.g., over pre-logit features from the penultimate layer. This allows for exploiting an enormous amount of engineering effort "for free" (e.g., network design).

Classification vs. Reconstruction. We note that most, if not all, GAN-based methods largely rely on the reconstruction error for open-set discrimination [47, 44, 39, 56, 35]. The underlying assumption is that closed-set data produces lower reconstruction error than the open-set. While this seems reasonable, it is challenging to reconstruct complex, high-resolution images [2, 7], like Cityscapes images shown in Fig. 2. On the contrary, for open-set discrimination, we directly use the discriminator as the open-set likelihood function. While this has been used as a baseline which does not work well in the literature [47, 44, 39, 56, 30], to the best of our knowledge, it is the first time that GAN-discriminator outperforms prior art on various benchmarks, thanks to the model selection via open validation (Section 3.1).

4. Experiment

We conduct extensive experiments to validate OpenGAN under various setups, and justify the advantage of exploiting OTS features and using the GAN-discriminator as the open-set likelihood function. We first briefly introduce three experimental setups below (details in later sections).

- *Setup-I* open-set discrimination splits a *single dataset* into open and closed sets w.r.t class labels, e.g., define MNIST digits 0-5 as the closed-set for training, and digits 6-9 as the open-set in testing. Although small-scale, this is a common experimental protocol for open-set discrimination that classifies open-vs-closed test examples [33, 35, 38, 57].
- *Setup-II* open-set recognition requires both K-way classification on the closed-set and open-set discrimination. We follow a "less biased" protocol [48] that constructs the open train&test-sets with *cross-dataset* images [51].
- *Setup-III* examines the open-set discrimination at pixel level in semantic segmentation, which evaluates pixel-level open-vs-closed classification accuracy [6, 23].

Implementation. We describe how to train the closed-world K-way classification networks which compute OTS features used for training OpenGAN^{fea} (Fig. 1d) and other methods (e.g., OpenMax [5] and C2AE [35]). For training K-way networks under *Setup-I* and *II*, we train a ResNet18 model [22] exclusively on the closed-train-set (with K-way cross-entropy loss). Under *Setup-III*, we use HRNet [52] as an OTS network, which is a top ranked model for semantic segmentation on Cityscapes [11]. We choose the penultimate/pre-logit layer of each K-way network to extract OTS features. Other layers also apply but we do not explore them in this work. Over the features, we train OpenGAN^{fea} discriminator (2MB), as well as the generator (2MB), with a multi-layer perceptron architecture. For comparison, we also train a ground-up OpenGAN^{pix} over pixels with a CNN architecture (∼14MB) [58]. We train our OpenGAN models using GAN techniques [40]. Compared to the segmentation network HRNet (250MB), OpenGAN^{fea} is quite lightweight that induces minimal compute overhead. We conduct experiments with PyTorch [36] on a single Titan X GPU. Code is available at https://github.com/aimerykong/OpenGAN

Evaluation Metric. To evaluate open-set discrimination that measures the open-vs-closed binary classification performance, we follow the literature [28, 35] and use the area under ROC curve (AUROC) [12]. AUROC is a calibration-free and threshold-less metric, simplifying comparisons between methods and reliable in large open-closed imbalance situation. For open-set recognition that measures (K+1)-way classification accuracy (K closed-set classes plus the (K+1)th open-set class), we report the macro average F1-score over all the (K+1) classes on the valsets [45, 5].

4.1. Compared Methods

We compare the following representative baselines and state-of-the-art methods for open-set recognition.

Baselines. First, We explore classic generative models learned on closed-train-set, including Nearest Neighbors (NNs) [41] and Gaussian Mixture Models (GMMs) which were found to perform quite well over L2-normalized OTS features [26]. We refer the reader to the supplemental for details of GMMs as they are strong yet underexplored baseline in the literature. Both models can be used for open-set discrimination by thresholding NN distances or likelihoods. We further examine the idea of outlier exposure [25] that learns an open-vs-closed binary classifier (CLS). Lastly, following classic work in semantic segmentation [16], we evaluate a (K+1)-way classifier trained with outlier exposure, in which we use the softmax score corresponding to the (K+1)th "other" class as the open-set likelihood.

Likelihoods. Many methods compute open-set likelihood on OTS features, including Max Softmax Probability (MSP) [24] and Entropy [49] (derived from softmax probabilities), and calibrated MSP (MSP$_c$) [29]. OpenMax [5] fits logits to Weibull distributions [46] that recalibrate softmax outputs for open-set recognition. C2AE [35] learns an additional K-way classifier over the OTS features using reconstruction errors, which are then used as the open-set likelihoods. GDM [28] learns a Gaussian Discriminant Model on OTS features and computes open-set likelihood

Table 1: **Open-set discrimination (Setup-I)** measured by area under ROC curve (AUROC)↑. We report methods marked by * with their best reported numbers in the compared papers. Recall that OpenGAN-0 does not train on outlier data (i.e., λ_0=0 in Eq. 2) and only selects discriminator checkpoints on the validation set. OpenGAN-0^{fea} clearly performs the best. Defined on the off-the-shelf (OTS) features of closed-world K-way networks, NN^{fea} and OpenGAN-0^{fea} work much better than their pixel version (NN^{pix} and OpenGAN-0^{pix}).

Dataset	MSP [24]	MSP$_c$ [29]	MCdrop [17]	GDM [28]	OpenMax [5]	GOpenMax [18]*	OSRCI [33]*	C2AE [35]*	CROSR [54]*	RPL [10]*	Hybrid [57]*	GDFR [37]*	NN^{pix} [41]	NN^{fea} [41]	OpenGAN -0^{pix}	OpenGAN -0^{fea}
MNIST	.977	.985	.984	.989	.981	.984	.988	.989	.991	.996	.995	—	.931	.981	.987	**.999**
SVHN	.886	.891	.884	.866	.894	.896	.910	.922	.899	.968	.947	.935	.534	.888	.881	**.988**
CIFAR	.757	.808	.732	.752	.811	.675	.699	.895	.883	.901	.950	.807	.544	.801	.971	**.973**
TinyImgNet	.577	.713	.675	.712	.576	.580	.586	.748	.589	.809	.793	.608	.528	.692	.795	**.907**

based on Mahalanobis distance.

Bayesian Networks. Bayesian neural networks estimate uncertainties via Monte Carlo estimates (MCdrop) [17, 31]. The estimated uncertainties are used as open-set likelihoods. We implement MCdrop via 500 samples.

GANs. GOpenMax [18] and OSRCI [33] train GANs to generate fake images to augment closed-set data for open-set recognition. Other types of GANs can also be used for open-set recognition, such as BiGANs [56], on which we show our technical insights (e.g., training on OTS features and directly using the discriminator) also apply (Table 2).

When possible, we train the methods using their open-source code. We implement NN, CLS and OpenGAN on both RGB images (marked with ^{pix}) and OTS features (marked with ^{fea}) for comparison. For fair comparison, we tune all the models for all methods on the same val-sets.

4.2. Setup-I: Open-Set Discrimination

Datasets. MNIST/CIFAR/SVHN/TinyImageNet are widely used in the open-set literature, and we follow the literature to experiment with these datasets [33, 35]. For each of the first three datasets that have ten classes, we randomly split 6 (4) classes of train/val-sets as the closed (open) train/val-sets respectively. For TinyImageNet that has 200 classes, we randomly split 20 (180) classes of train/val-sets as the closed (open) train/val-set. On each dataset and for each method, we repeat five times with different random splits and report the average AUROC on the val-set [33, 35]. As all methods have small standard deviations in their performance (<0.02), we omit them for brevity.

Results. As this setup assumes no open training data, we cannot train discriminative classifiers like CLS. But we can still train OpenGAN-0 that uses GAN-discriminator (with model selection) as the open likelihood function. We have two salient conclusions from the results in Table 1. (1) Methods (e.g., NN and OpenGAN) work better on OTS features than pixels, suggesting that OTS features computed by the underlying K-way network are already good representations for open-set recognition. (2) OpenGAN-0^{fea} performs the best and OpenGAN-0^{pix} is competitive with prior methods such as GDM and GMM, suggesting that the GAN-discriminator is a powerful open likelihood function.

Further Analysis. There are many other GAN-based open-set methods, such as training BiGANs [47, 55, 56] or adversarial autoencoders [39, 44] on raw images, and using the reconstruction error as open-set likelihood [47, 44, 56, 1, 13]. We show our technical insights apply to different GAN architectures for open-set recognition: (1) using GAN-discriminator as the open-set likelihood function instead of pixel reconstruction errors, and (2) training them on OTS features rather than raw pixels. We hereby analyze a typical BiGAN-based method [56], which learns a BiGAN with both the reconstruction error and the GAN-discriminator. We compare BiGAN's performance by either using the reconstruction error (BiGAN$_r$) or its discriminator (BiGAN$_d$) for open-set recognition. We also compare building BiGANs on either pixels (BiGAN^{pix}) or features (BiGAN^{fea}). Table 2 lists detailed comparisons under *Setup-I* (all models are selected on the val-sets). Clearly, our conclusions hold regardless of the base GAN architecture: 1) using OTS features rather than pixels (cf. BiGAN^{fea} vs BiGAN^{pix}), and 2) more importantly, using discriminators instead of reconstruction errors (cf. BiGAN$_d$ vs. BiGAN$_r$).

Table 2: Our technical insights apply to other GAN-based open-set discrimination methods: 1) using BiGAN-discriminator as the open likelihood function works better than using reconstruction errors (BiGAN$_d^{fea}$ vs. BiGAN$_r^{fea}$), and 2) learning BiGANs on OTS features works much better than pixels (BiGAN$_d^{fea}$ vs. BiGAN$_d^{pix}$). The results are comparable to Table 1.

dataset	BiGAN$_r^{pix}$	BiGAN$_r^{fea}$	BiGAN$_d^{pix}$	BiGAN$_d^{fea}$
MNIST	.976	.998	.986	.999
SVHN	.822	.976	.880	.993
CIFAR	.924	.967	.968	.973

4.3. Setup-II: Cross-Dataset Open-Set Recognition

Using cross-dataset examples as the open-set is another established protocol [29, 28, 25, 15]. We follow the "less biased" protocol introduced in [48], which uses three datasets for benchmarking that reduces dataset-level bias [51]. This protocol tests the generalization of open-set methods to diverse open testing examples.

Datasets. We use TinyImageNet as the closed-set for K-way classification (K=200). Images of each class are split into 500/50/50 images as the train/val/test sets. Following [48], we construct open train/val and test sets using different datasets [51], including MNIST (MN), SVHN (SV), CIFAR (CF) and Cityscapes (CS). For example, we use MNIST *train*-set to tune/train a model, and test it on CI-

Table 3: **Open-set recognition (Setup-II)** measured by AUROC↑, and macro-averaged F1-score↑ over all (K+1) classes. We use TinyImageNet (K=200) as the closed-set, and four different datasets as the open-sets. To report a method on a specific open-test-set out of four (first column), we perform four runs in which we use one of the four datasets as a validation set for training/tuning, and then average the performance measures over the four runs with a superscript marking the standard deviation. Methods such as Nearest Neighbor (NN) do not need tuning and hence have zero deviations. We provide a summary number in the bottom macro row by averaging the results over all open-test-sets. Detailed results in Table 4. Clearly, a binary classifier trained on features (CLS^{fea}) already outperforms prior methods. However, when trained on pixels, CLS^{pix} works poorly in AUROC due to overfitting to high-dimensional raw images, but performs decently in F1. To note, without handling the open-set, the K-way model (trained only on the closed-set TinyImageNet) achieves 0.553 F1-score over (K+1) classes, suggesting that, when K is large (K=200 here), F1-score can hardly reflect open-set discrimination performance which is better measured by AUROC. While largely underexplored in the literature, training a (K+1)-way model works quite well. Clearly, OpenGAN^{fea} works the best in both AUROC and F1-score. Please refer to Fig. 3(f-i) for ROC curves, and F1-scores vs. thresholds on the open-set likelihood.

		MSP	OpenMax	NN^{fea}	GMM	C2AE	MSP$_c$	MCdrop	GDM	CLS^{pix}	(K+1)	CLS^{fea}	Open	Open
open-test	*metric*	[24]	[5]	[41]	[26]	[35]	[29]	[17]	[28]				GAN^{pix}	GAN^{fea}
CIFAR	AUROC	.769$^{.000}$	.669$^{.011}$	.927$^{.000}$	.961$^{.013}$	.767$^{.020}$	.791$^{.007}$	.809$^{.005}$	.961$^{.007}$	.754$^{.367}$	.880$^{.091}$	.928$^{.113}$	.981$^{.027}$	.980$^{.011}$
	F1	.548$^{.002}$	.507$^{.001}$	.525$^{.000}$	.544$^{.002}$	.564$^{.002}$	.553$^{.003}$	.564$^{.001}$	.519$^{.003}$	.545$^{.032}$	.558$^{.017}$	.555$^{.027}$	.563$^{.035}$	.585$^{.003}$
SVHN	AUROC	.695$^{.000}$	.691$^{.014}$	.994$^{.000}$	.990$^{.016}$	.657$^{.018}$	.863$^{.013}$	.783$^{.009}$	.999$^{.006}$	.701$^{.224}$	.948^{068}	.955$^{.052}$	.980$^{.014}$	.991$^{.013}$
	F1	.567$^{.002}$	.551$^{.002}$	.545$^{.000}$	.574$^{.002}$	.565$^{.001}$	.572$^{.002}$	.572$^{.001}$	.575$^{.002}$	.572$^{.027}$	.564$^{.015}$	.578$^{.014}$	.574$^{.009}$	.583$^{.008}$
MNIST	AUROC	.764$^{.000}$	.690$^{.019}$	.901$^{.000}$	.964$^{.021}$	.755$^{.008}$	.832$^{.017}$	.801$^{.009}$	.957$^{.007}$	.986$^{.327}$	.944$^{.015}$	.961$^{.083}$	.983$^{.068}$	.989$^{.014}$
	F1	.559$^{.001}$	.536$^{.013}$	.553$^{.000}$	.547$^{.008}$	.575$^{.001}$	.564$^{.001}$	.563$^{.001}$	.552$^{.002}$	.565$^{.020}$	.586$^{.021}$	.583$^{.010}$	.569$^{.016}$	.582$^{.005}$
Citysc.	AUROC	.789$^{.000}$	.693$^{.021}$	.715$^{.000}$	.867$^{.016}$	.814$^{.010}$	.851$^{.003}$	.868$^{.003}$	.513$^{.005}$	.646$^{.332}$	.971$^{.050}$	.828$^{.032}$	.933$^{.026}$	.978$^{.013}$
	F1	.579$^{.002}$	.514$^{.002}$	.583$^{.000}$	.572$^{.003}$	.589$^{.002}$	.583$^{.001}$	.571$^{.001}$	.546$^{.003}$	.589$^{.007}$	.561$^{.029}$	.587$^{.006}$	.588$^{.007}$	.587$^{.000}$
average	AUROC	.754	.686	.884	.945	.748	.834	.815	.857	.772	.936	.918	.969	**.984**
	F1	.560	.527	.552	.559	.569	.568	.567	.548	.568	.565	.576	.573	**.584**

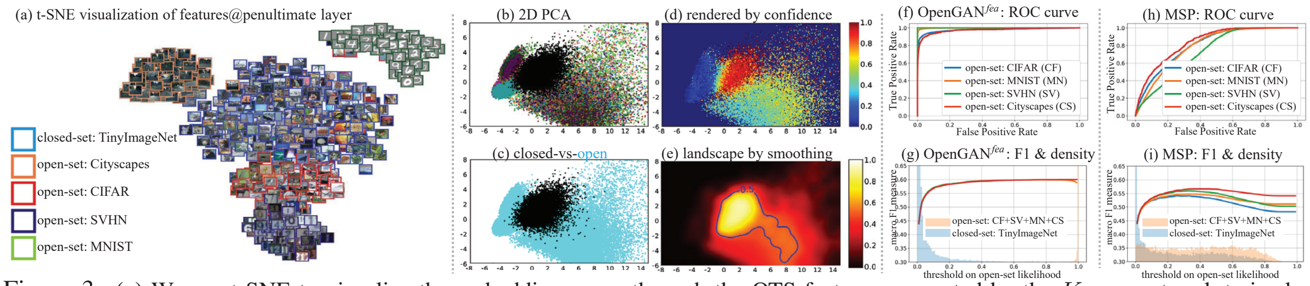

Figure 3: **(a)** We use t-SNE to visualize the embedding space through the OTS features computed by the K-way network trained on TinyImageNet train-set. Images from the other datasets are open-set examples. Clearly, closed and open examples are well separated in the feature space. We further visualize the "landscape" of the OpenGAN^{fea} open-set discriminator, by **(b)** projecting the OTS features into 2D using PCA; **(c)** coloring them with their closed/open labels; **(d)** rendering them with their open-set likelihoods computed by OpenGAN^{fea}; **(e)** smoothing with Gaussian Filtering overlaid with the OpenGAN's decision boundary. We further compare OpenGAN^{fea} (tuned on SVHN) and MSP in **(f-g)** for open-set discrimination by ROC curves, and in **(h-i)** for open-set recognition by curves of the F1-score vs. thresholds of open likelihold. We render the density of open and closed testing data using shadows in **(g)** and **(i)**. In these plots, we use each of the four cross-dataset open-test-sets (unseen in training) as an independent open-set to draw the curves. The curves clearly show that OpenGAN significantly outperforms MSP on open-set discrimination (AUROC) and open-set recognition (F1).

FAR *test*-set as open-test set. This allows for analyzing how open-set methods generalize to diverse open testing examples (cf. Table 4). We use bilinear interpolation to resize all images to 64x64 to match TinyImageNet image resolution.

Results. Table 3 shows detailed results. First, methods perform much better on features than pixels (e.g., CLS^{fea} vs. CLS^{pix}); and our OpenGAN performs the best. Perhaps surprisingly, OpenMax, a classic open-set, does not work well in this setup. This is consistent with the results in [15, 48]. We conjecture that OpenMax cannot effectively recognize cross-dataset open-set examples represented by logit features (computed by the K-way network) which are too invariant to be adequate for open-set recognition. Moreover, the (K+1)-way classifier also works quite well, even outperforming the open-vs-closed binary classifiers (CLS) in AUROC. Next we analyze why the binary classifier CLS (as widely done since [25]) are less effective.

Further Analysis. Table 4 lists detailed results of OpenGAN, CLS (λ_G=0 in Eq. 2) and OpenGAN-0 (λ_o=0 in Eq. 2), when trained/tuned and tested on different cross-dataset open-set examples. All methods perform better on OTS features than pixels (cf. CLS^{fea} vs. CLS^{pix}); and work almost perfectly when trained and tested with the same open-set dataset, e.g., column-cf under "CIFAR-train (cf)" where we use CIFAR images as the open-set data. However, when tested on a different dataset of open-set examples, CLS performs quite poorly (especially when built on pixels) because it overfits easily to high-

Table 4: **Diagnostic analysis for cross-dataset open-set discrimination** measured by AUROC↑. In this setup, the TinyImageNet train/val/test sets serve as the closed train/val/test sets, and open train/test sets are the other two different datasets. Following outlier exposure [24], we train/tune CLS and OpenGAN on a cross-dataset as the open train-set. Recall that we do not train OpenGAN-0 on *any* open examples, although we tune it on the respective cross-dataset open train-set. CLS and OpenGAN use their last-epoch checkpoints to report performance. For better comparison, we report the average AUROC performance across all open-val-sets in the last column. We color the entries that have AUROC <0.9 with red, implying these models overfit to the open-train-set and generalize poorly on the other open-test-set. OpenGAN^{fea} clearly performs the best; while CLS (esp. CLS^{pix} which operates on pixels) generalizes poorly. Perhaps surprisingly, OpenGAN-0 performs equally well although it does not train on open taining data.

open-val-set	CIFAR10 (CF)				SVHN (SV)				MNIST (MN)				Cityscapes (CS)				*avg.*
open-test-set	CF	SV	MN	CS	CF	SV	MN	CS	CF	SV	MN	CS	CF	SV	MN	CS	
CLS^{pix}	.999	.999	.101	.895	.935	.999	.453	.972	.411	.340	.999	.113	.317	.512	.100	.999	.634
OpenGAN-0^{pix}	.999	.998	.550	.999	.999	.999	.993	.999	.999	.968	.999	.911	.999	.999	.915	.999	.958
OpenGAN^{pix}	.999	.999	.989	.933	.974	.999	.997	.967	.976	.998	.999	.835	.967	.928	.950	.999	.969
CLS^{fea}	.999	.933	.916	.699	.940	.999	.979	.863	.893	.961	.999	.781	.881	.926	.949	.968	.918
OpenGAN-0^{fea}	.999	.998	.997	.999	.964	.996	.996	.946	.952	.992	.994	.934	.994	.995	.992	.997	**.984**
OpenGAN^{fea}	.999	.999	.990	.973	.974	.999	.996	.971	.976	.998	.999	.967	.973	.968	.970	.999	**.984**

dimensional pixel images [48]. In contrast, with *fake* open-data generated adversarially, OpenGAN and its special form OpenGAN-0 perform and generalize much better. Nevertheless, this implies a failure mode of OpenGAN, because the open-set data used in training could be quite different from those in testing, potentially leading to an OpenGAN that perform poorly in the real open world. Perhaps surprisingly, OpenGAN-0^{fea} performs as well as OpenGAN^{fea}, although it does not train on open-set data. This further shows the merit of generating *fake* open examples to augment heavily-biased open-set training data, and our technique insights (as previously analyzed under *Setup-I*): 1) using GAN-discriminator as the likelihood function, and 2) training GANs on OTS features rather than pixels.

Visualization. Fig. 1 shows some synthesized images by GAN^{pix}, and we visualize more in the supplement. To intuitively illustrate how the synthesized images help better span the open-world, we analyze why a simple discriminator works so well when trained on the OTS features. We visualize the features in Fig. 3 (a) and "decision landscape" in Fig. 3 (b-e), demonstrating that the closed- and open-set images are clearly separated in the feature space.

4.4. Setup-III: Open-Set Semantic Segmentation

Open-set semantic segmentation has been explored in recent work [6, 23], which creates synthetic open-set pixels by pasting virtual objects (e.g., cropped from PASCAL VOC masks [16]) on Cityscapes images. In this work, we do not generate synthetic pixels but instead repurpose "other" pixels (outside the set of K classes) that already exist in Cityscapes. Interestingly, classic semantic segmentation benchmarks evaluate these "other" pixels as a separate background class [16], but Cityscapes ignores them in its evaluation (as do many other contemporary datasets [8, 3, 42, 34]). The historically-ignored pixels include vulnerable objects (e.g., strollers in Fig. 2), and can be naturally evaluated as open-set examples.

Datasets. Cityscapes [11] contains 1024x2048 high-resolution urban scene images with 19 class labels for semantic segmentation. We construct our train- and val-sets from its 2,975 training images, in which we use the last 10 images as val-set and the rest as train-set. We use its official 500 validation images as our test-set. The "other" pixels (cf. Fig. 2) are the open-set examples in this setup. We refer readers to the supplemental for details, such as model architecture, batch construction, weight tuning, etc.

Pixel Generation. As Cityscapes has high-resolution images (1024x2048), it is nontrivial to train OpenGAN^{pix}, especially its special form OpenGAN-0^{pix}, which must learn to generate high-resolution images. We find the successful training of OpenGAN-0^{pix} depends on the resolution of images to be generated: we train OpenGAN-0^{pix} by generating patches (64x64), not full-resolution images.

Results. Table 5 lists quantitative results. As we train OpenGAN and CLS with open pixels, we diagnose in Fig. 5 the open-set performance by varying the number of training images that provide the open-training pixels, along with closed-training pixels from all training images. First, these results show that OpenGAN^{fea} substantially outperforms all other methods. Generally speaking, the methods that process features outperform those that process pixels (e.g., OpenGAN and CLS in Fig. 5). This suggests that OTS features (from the segmentation network) serve as a powerful representation for open-set pixel recognition. The curves in Fig. 5 imply that methods with enough data on pixels should work (e.g., achieving similar performance as on features). This is consistent with evidence from semantic segmentation works. However, methods saturate more quickly on OTS features than pixels, suggesting the benefit of using OTS features for open-set recognition. Moreover, OpenGAN-0 performs better than CLS when trained on fewer open-training images (e.g., 10). But with modest number of open training images (e.g., 50), CLS outperforms OpenGAN-0 and other classic methods (e.g., OpenMax and

Table 5: **Comparison in open-set semantic segmentation** on Cityscapes (AUROC ↑). All methods are implemented on top of the segmentation network HRNet [52] except the ones operating on pixels (as marked by ^{pix}). Our approach OpenGAN^{fea} clearly performs the best. Fig. 5 analyzes OpenGAN trained with varied number of open-set pixels, when built on either pixels or OTS features.

MSP [24]	Entropy [49]	OpenMax [5]	C2AE [35]	MSP$_c$ [29]	MCdrop [17]	GDM [28]	GMM [26]	HRNet-(K+1)	OpenGAN-0^{fea}	CLS^{fea}	OpenGAN^{fea}
.721	.697	.751	.722	.755	.767	.743	.765	.755	.709	.861	**.885**

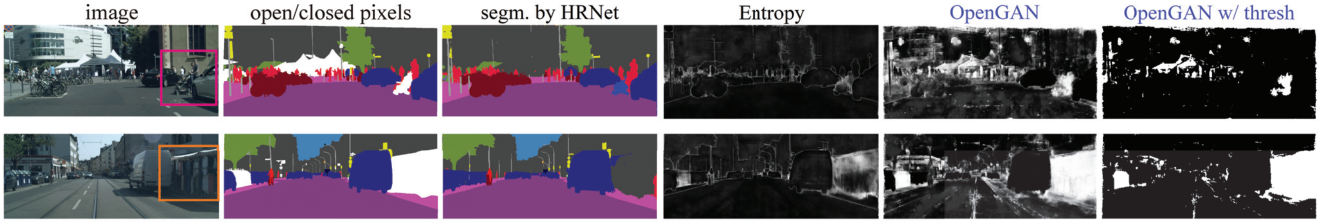

Figure 4: Qualitative results of two testing images, on which a state-of-the-art network (HRNet) misclassifies the *unknown* categories stroller/street-shop as motorcycle/building. From left to right of each row: the input image, its per-pixel semantic labels (in which `white` regions are open-set pixels), the semantic segmentation result by HRNet, open-set likelihoods by Entropy, our OpenGAN^{fea}, and its thresholded open-pixel map (threshold=0.7). OpenGAN clearly captures most open-set pixels (the white ones). Note that the street-shop is a real open-set example because Cityscapes train-set does not have another street-shop like this size and content (i.e., selling clothes).

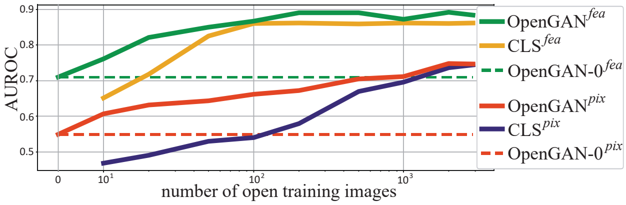

Figure 5: Diagnostic study w.r.t AUROC vs. number of open images which provide open-set training pixels. Our methods perform better on OTS features than pixels. Recall OpenGAN-0 is equivalent to training a normal GAN (without open training data) and using its discriminator as open-set likelihood. With some open training data (e.g., 100 open images), CLS outperforms OpenGAN-0; but OpenGAN consistently performs the best.

C2AE in Table 5) which assume no open training data. This confirms the effectiveness of Outlier Exposure, even with a modest amount of outliers [25].

Bayesian networks (MCdrop and MSP$_c$) outperform the baseline MSP, showing that uncertainties can be reasonably used for open-set recognition. Lastly, we train a "groundup" (K+1)-way HRNet model that treats "other" pixels as the (K+1)th background class [16], shown by HRNet-(K+1) in Table 5. It performs better than other typical openset methods but much lower than the simple open-vs-closed binary classifier CLS^{fea}, presumably because the (K+1)-way model has to strike a balance over all the (K+1) classes while the binary CLS benefits from training on more balanced batches of closed/open pixels.

Visualization. Fig. 4 qualitatively compares OpenGAN and the entropy method (more visual results are in the supplemental). The visualization shows OpenGAN sufficiently recognize open-set pixels. It also implies failure happens when OpenGAN misclassifies open-vs-closed pixels. Fig. 6 compares some generated patches by OpenGAN-0^{fea} and OpenGAN-0^{fea}, intuitively showing why using OTS features leads to better performance for open-set recognition.

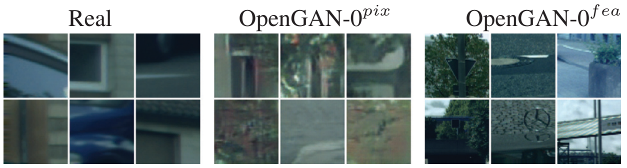

Figure 6: Visuals of Cityscapes real image patches (left), synthesized patches by OpenGAN-0^{pix} (mid) and OpenGAN-0^{fea} (right). As OpenGAN-0^{fea} generates features instead of pixel patches, we "synthesize" the patches analytically – for a generated feature, from training pixels represented as OTS features, we find the nearest-neighbor pixel feature (w.r.t L1 distance), and use the RGB patch centered at that pixel as the "synthesized" patch. We can see OpenGAN-0^{pix} synthesizes realistic patches w.r.t color and tone, but it (0.549 AUROC) notably underperforms OpenGAN-0^{fea} (0.709 AUROC) for open-set segmentation. The "synthesized" patches by OpenGAN-0^{fea} capture many open-set objects, such as bridge, back-of-traffic-sign and unknown-static-objects, none of which belong to any of the 19 closed-set classes in the Cityscapes benchmark. This intuitively shows why methods work better on OTS features than pixels.

5. Conclusion

We propose **OpenGAN** for open-set recognition by incorporating two technical insights, 1) training an open-vs-closed classifier on OTS features rather than pixels, and 2) adversarialy synthesizing *fake* open data to augment the set of open-training data. With OpenGAN, we show using GAN-discriminator *does* achieve the state-of-the-art on open-set discrimination, once being selected using a val-set of real outlier examples. This is effective even when the outlier validation examples are sparsely sampled or strongly biased. OpenGAN significantly outperforms prior art on both open-set image recognition and semantic segmentation.

Acknowledgement. This work was supported by the CMU Argo AI Center for Autonomous Vehicle Research.

References

[1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision (ACCV)*, 2018. 2, 5

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 3, 4

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 7

[4] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*, 2015. 3

[5] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, 2016. 1, 2, 4, 5, 6, 8

[6] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The fishyscapes benchmark: Measuring blind spots in semantic segmentation. *arXiv preprint arXiv:1904.03215*, 2019. 4, 7

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 3, 4

[8] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 7

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009. 2

[10] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *ECCV*, 2020. 3, 5

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 4, 7

[12] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML*, 2006. 4

[13] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. In *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2018. 2, 5

[14] David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. In *ICLR*, 2020. 2

[15] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. In *NeurIPS*, 2018. 1, 2, 3, 5, 6

[16] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 4, 7, 8

[17] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 2, 5, 6, 8

[18] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. In *British Machine Vision Conference (BMVC)*, 2017. 2, 3, 5

[19] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[20] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *ICCV*, 2019. 2

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2, 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[23] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A benchmark for anomaly segmentation. *arXiv preprint arXiv:1911.11132*, 2019. 4, 7

[24] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 2, 4, 5, 6, 7, 8

[25] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019. 1, 2, 3, 4, 5, 6, 8

[26] Shu Kong and Deva Ramanan. An empirical exploration of open-set recognition via lightweight statistical pipelines, 2021. 4, 6, 8

[27] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018. 3

[28] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018. 2, 4, 5, 6, 8

[29] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018. 1, 2, 4, 5, 6, 8

[30] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1517–1528, 2019. 1, 2, 4

[31] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 2020. 5

[32] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018. 3

[33] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *ECCV*, 2018. 2, 3, 4, 5

[34] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 7

[35] Poojan Oza and Vishal M. Patel. C2AE: class conditioned auto-encoder for open-set recognition. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 8

[36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 4

[37] Pramuditha Perera, Vlad I Morariu, Rajiv Jain, Varun Manjunatha, Curtis Wigington, Vicente Ordonez, and Vishal M Patel. Generative-discriminative feature representations for open-set recognition. In *CVPR*, 2020. 5

[38] Pramuditha Perera and Vishal M Patel. Deep transfer learning for multiple class novelty detection. In *CVPR*, 2019. 4

[39] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *NeurIPS*, 2018. 1, 2, 4, 5

[40] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 4

[41] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2000. 4, 5, 6

[42] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 7

[43] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *ICLR*, 2020. 2

[44] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *CVPR*, 2018. 1, 2, 4, 5

[45] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2012. 1, 2, 3, 4

[46] Walter J Scheirer, Anderson Rocha, Ross J Micheals, and Terrance E Boult. Meta-recognition: The theory and practice of recognition score analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 4

[47] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017. 1, 2, 4, 5

[48] Alireza Shafaei, Mark Schmidt, and James J. Little. A less biased evaluation of out-of-distribution sample detectors. In *British Machine Vision Conference (BMVC)*, 2019. 1, 2, 3, 4, 5, 6, 7

[49] Jacob Steinhardt and Percy S Liang. Unsupervised risk estimation using only conditional independence structure. In *NeurIPS*, 2016. 4, 8

[50] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *CVPR*, 2020. 2

[51] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011. 4, 5

[52] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2, 4, 8

[53] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *ICCV*, pages 1511–1519, 2015. 2

[54] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *CVPR*, 2019. 1, 2, 5

[55] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018. 5

[56] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. Adversarially learned anomaly detection. In *IEEE International Conference on Data Mining (ICDM)*, 2018. 1, 2, 4, 5

[57] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *ECCV*, 2020. 2, 4, 5

[58] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 4

[59] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018. 1, 2