OpenGAN: Open-Set Recognition via Open Data Generation

Shu Kong, and Deva Ramanan

Abstract—Real-world machine learning systems need to analyze test data that may differ from 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, 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 are unlikely to exhaustively span the open-world. The latter does not work well, presumably due to the unstable 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. This allows OpenGAN to be implemented via a lightweight discriminator head built on top of an existing K-way network. Extensive experiments show that OpenGAN significantly outperforms prior open-set methods.

Index Terms—Open-Set Recognition, Outlier, Out-of-Distribution Detection, Generative Adversarial Network, Discriminative Adversarial Network, Open-World, Visual Perception, Semantic Segmentation.

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 never) observed during training. Fig. 1 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 [3], [4], and out-of-distribution detection [5]. In K-way classification, this task 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 [6]. Typically, open-set discrimination assumes no examples from the "other" class available during training [7], [8], [9]. In this setup, one elegant approach is to learn the closed-set data distribution with a GAN, making use of the GAN-discriminator as the open-set likelihood function (Fig. 2b) [10], [11], [12], [13], [14]. However, this does not work well due to the unstable training of GANs. Recent work has shown that outlier exposure (Fig. 2a), 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 [4], [15]. However, such discriminators fail to generalize to diverse open-set data [16] because they overfit to the available set of training outliers, which are often biased and fail to exhaustively span the open-world.

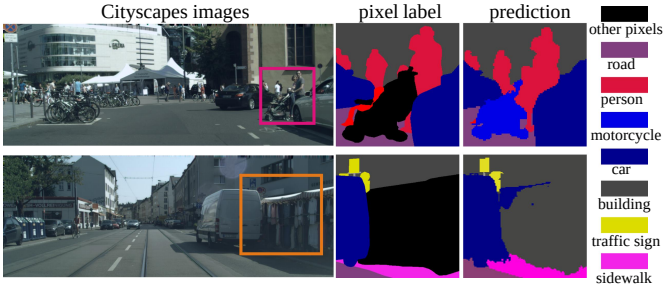
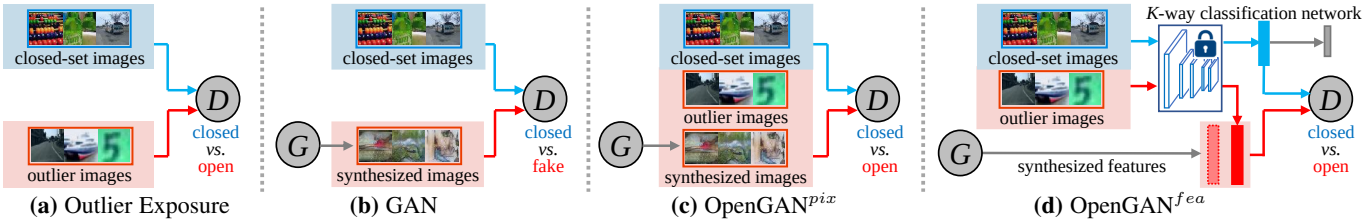


Fig. 1: We motivate open-set recognition with safety concerns in autonomous vehicles (AVs). Contemporary benchmarks such as Cityscapes [1] focus on K classes of interest for evaluation, ignoring a sizeable set of "other" pixels that include vulnerable objects like wheelchairs and strollers (upper row). As a result, most state-of-the-art segmentation models [2] also ignore these pixels during training, resulting in a stroller misclassified as a "motorcycle" (top) and a street-market misclassified as a "building". Such misclassifications may be critical for AVs because these objects may require different plans for obstacle avoidance (e.g., "yield" or "slow-down"). Fig. 6 shows our approach, which explicitly augments state-of-the-art segmentation models with open-set reasoning.

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. 2c). 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. 2d). We find such discriminators generalize much better.

• *Work was done when Shu Kong was with the Robotics Institutes, Carnegie Mellon University. Shu Kong is now affiliated with the Department of Computer Science & Engineering, Texas A&M University; Deva Ramanan is affiliated with both the Robotics Institutes at Carnegie Mellon University and Argo AI. E-mail: shu@tamu.edu, deva@cs.cmu.edu*



(a) Outlier Exposure **(b)** GAN **(c)** OpenGAN^{pix} **(d)** OpenGAN^{fea}

Fig. 2: We explore open-set recognition, which requires the ability to discriminate open-set test examples outside K classes of interest. **(a)** A discriminative method, Outlier Exposure [4], exploits 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-set data [16]. **(b)** To better span the open-world, one idea is to synthesize *fake* outliers using a generator that is trained to fool the binary discriminator D. This is exactly the Generative Adversarial Network (GAN) architecture [17]. Because GAN is designed to learn the closed-set data distribution, we can use the GAN discriminator as the open-set likelihood function. However, it does not work well due to the unstable training of GANs [10], [11], [12], [13], [14]. **(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) low-dimensional *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.

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 [4]). (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 functions.

2 Related Work

Open-Set Recognition. There are multiple lines of work addressing open-set discrimination, such as anomaly detection [3], [5], [18], outlier detection [11], [12], and open-set recognition [6], [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 first train a closed-world K-way classification network on the closed-set [7], [20] and then exploit the trained network for open-set discrimination [6], [9], [21]. 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 [8], [22], [23], [24]. To recognize open-set examples, they resort to post-hoc functions like density estimation [3], [25], uncertainty modeling [5], [26], and input image reconstruction [12], [24], [27], [28]. 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 [17], conceptually, a GAN-discriminator trained on the closed-set naturally serves as an open-set likelihood function. However, this does not work well [10], [11], [12], [13], [14], 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 [10], [11], [29], [30], [31]. 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. [4], [15], [32] 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 [16]. 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 the recognized closed-set [6], [9], [33]. The core and challenging problem to open-set recognition is the first step, i.e., open-set discrimination, because it typically assumes that open-set examples are not available during training [9], [23]. However, [4], [15] demonstrate that outlier exposure, or the ability to train on *some* outlier examples, can greatly improve open-set discrimination via the training of a simple discriminative open-vs-closed binary classifier (Fig. 2a). 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 [16]. 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. We refer the reader to Fig. 2 for the nutshell of our methodology.

3.1 Methodology

Let x be a data example, represented by either 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 $D(x) = p(y\text{=}\text{“closed-set”}|x)$. 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 [4], but underperforms when they fail to span the open-world [16].

Synthetic Open Data. One solution to the above is to exploit synthetic training data, which might improve the generalizability of classifier D. 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 [34], [35]. A natural solution is to adversarially train the generator G to produce difficult examples that fool the classifier D using a GAN loss:

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

It is worth noting that a *perfectly* trained generator G would generate realistic closed-set images that eventually make 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- and closed-set data and the *fake* open-data into a single (GAN-like) minimax optimization over D and G:

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

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 past work suggests that GAN-discriminators do not work well as open-set likelihood functions, we show they *do* achieve the state-of-the-art *once* selected with 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 [17]. 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. This is 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 appendix). 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 test-time 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 Comparison to Prior GAN Methods

We compare OpenGAN to a number of prior methods that use GANs for open-set discrimination.

Discriminator vs. Generator. Typically, GANs aim at generating realistic images [36], [37]. As a result, prior work in open-set recognition has focused on using GANs to generate realistic open-training images [22], [23], [38]. 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 [22], [23], [38]. 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 [4], [12], [13]. However, generating high-dimensional realistic images is challenging per se [36], [37] and may not be necessary to open-set recognition [12]. As such, we build GANs over OTS feature embeddings learned by the closed-world K-way classification network, e.g., over pre-logit features at 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 [9], [10], [11], [12], [13]. The underlying assumption is that closed-set data produces lower reconstruction error than the open-set. While this is reasonable just like building generative models over closed-set data [39], it is challenging to reconstruct complex, high-resolution images [36], [37], such as the Cityscapes images shown in Fig. 1. On the contrary, we directly use the discriminator as the open-set likelihood function. Such a baseline has been shown to perform poorly in large body of prior work [10], [11], [12], [13], [14]. Importantly, to the best of our knowledge, ours is the first result to demonstrate the strong performance of GAN-discriminators, thanks in large part to 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 [9], [23], [25], [42].
- *Setup-II* open-set recognition requires both K-way classification on the closed-set and open-set discrimination. We follow a

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 [20]	MSP_c [5]	MCdrop [26]	GDM [21]	OpenMax [7]	GOpenMax [22]*	OSRCI [23]*	GMM [39]	C2AE [9]*	CROSR [8]*	CGDL [24]	RPL [35]*	Hybrid [25]*	GDFR [40]*	NN^{pix} [41]	NN^{fea} [41]	OpenGAN -0^{pix}	OpenGAN -0^{fea}
MNIST	.977	.985	.984	.989	.981	.984	.988	.993	.989	.991	.994	.996	.995	—	.931	.981	.987	**.999**
SVHN	.886	.891	.884	.866	.894	.896	.910	.914	.922	.899	.935	.968	.947	.935	.534	.888	.881	**.988**
CIFAR	.757	.808	.732	.752	.811	.675	.699	.817	.895	.883	.903	.901	.950	.807	.544	.801	.971	**.973**
TinyIN	.577	.713	.675	.712	.576	.580	.586	.817	.748	.589	—	.809	.793	.608	.528	.692	.795	**.907**

"less biased" protocol [16] that constructs the open train&test-sets with *cross-dataset* images [43].

- *Setup-III* examines the open-set discrimination at pixel level in semantic segmentation, which evaluates pixel-level open-vs-closed classification accuracy [44], [45].

Implementation. We describe how to train the closed-world K-way classification networks which compute OTS features used for training OpenGAN^{fea} (Fig. 2d) and other methods (e.g., OpenMax [7] and C2AE [9]). For training K-way networks under *Setup-I* and *II*, we train a ResNet18 model [46] exclusively on the closed-train-set (with K-way cross-entropy loss). Under *Setup-III*, we use HRNet [2] as an OTS network, which is a top ranked model for semantic segmentation on Cityscapes [1]. 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 ($\sim$14MB) [47]. We train our OpenGAN models using GAN techniques [48]. Compared to the segmentation network HR-Net (250MB), OpenGAN^{fea} is quite lightweight that induces minimal compute overhead. We refer the reader to Appendix B for network architectures. We conduct experiments with PyTorch [49] 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 [9], [21] and use the area under ROC curve (AUROC) [50]. 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 [6], [7].

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 the 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 yet underexplored in the literature of open-set recognition [39]. We refer the reader to the Appendix A for a brief overview how to learn these models. For open-set discrimination, we threshold NN distances or GMM likelihoods. We further examine the idea of outlier exposure [4] that learns an open-vs-closed binary classifier (CLS). Lastly, following classic work in semantic segmentation [51], 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) [20] and Entropy [52] (derived from softmax probabilities), and calibrated MSP (MSP_c) [5]. OpenMax [7] fits logits to Weibull distributions [53] that recalibrate softmax outputs for open-set recognition. C2AE [9] learns an additional K-way classifier over the OTS features using reconstruction errors, which are then used as the open-set likelihoods. GDM [21] learns a Gaussian Discriminant Model on OTS features and computes open-set likelihood based on Mahalanobis distance.

Bayesian Networks. Bayesian neural networks estimate uncertainties via Monte Carlo estimates (MCdrop) [26], [54]. The estimated uncertainties are used as open-set likelihoods. We implement MCdrop via 500 samples.

GANs. GOpenMax [22] and OSRCI [23] 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 [13], on which we show our technical insights (e.g., training on OTS features and directly using the discriminator) also apply (cf. 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 [9], [23]. 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 [9], [23]. 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 on the same val-set) 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 discrimination. (2) OpenGAN-0^{fea} performs the best and OpenGAN-0^{pix} is competitive with prior methods such as GDM

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

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 [10], [13], [55] or adversarial autoencoders [11], [12] on raw images, and using the reconstruction error as open-set likelihoods [10], [11], [13], [30], [31]. 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 [13], 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).

Visualization. Recall that OpenGAN-0^{pix} trains a generator that synthesizes fake open-set training images. We visualize the synthesized images in Fig. 3. Interestingly, patterns, colors and shapes in the synthesized images resemble those in the real closed-set images. However, the GAN-discriminator produces very low closed-set confidence scores for them, demonstrating that (1) these synthesized images are classified as the open-set, and (2) they are easy to recognize as fake / open-set. This also suggests that the synthesized data are insufficient to be used for model selection, which is studied further in Section 4.5.

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

Using cross-dataset examples as the open-set is another established protocol [4], [5], [15], [21]. We follow the "less biased" protocol introduced in [16], which uses three datasets for benchmarking to reduce dataset-level bias [43]. 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 [16], we construct open train/val and test sets using different datasets [43], including MNIST (MN), SVHN (SV), CIFAR (CF) and Cityscapes (CS). For example, we use MNIST *train*-set to tune/train a model and CIFAR *test*-set as the open-set for testing. This allows for analyzing how open-set methods generalize to diverse open testing examples (cf. Table 4). As different datasets have different resolutions of the images, 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 by learning to synthesize fake open-set training data. Perhaps surprisingly, OpenMax, a classic open-set, does not work well in this setup. This is consistent with the results in [15], [16]. 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 discrimination. Moreover, the (K+1)-way classifier works quite well, even better than the open-vs-closed binary classifiers (CLS) in AUROC.

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-dimensional pixel images [16]. 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 performs 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 open-set likelihood function, and (2) training GANs on OTS features rather than pixels.

Visualization. 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. 4 (a) and "decision landscape" in Fig. 4 (b-e), demonstrating that the closed- and open-set images are clearly separated in the feature space. Recall that OpenGAN-0^{pix} trains a generator that synthesizes fake open-set images. We visualize some of these fake images in Fig. 5. The generated images have similar colors and tones to the real closed-set ones but their contents are not real objects. This suggests that the synthesized fake open-set images are not closed-set but "real" outliers. However, from the low confidence scores of classifying the generated fake images as closed-set, we can see the discriminator easily recognizes them as open-set / outliers. This further shows that using synthesized data is inadequate for model selection.

4.4 Setup-III: Open-Set Semantic Segmentation

Open-set semantic segmentation has been explored recently [44], [45], which creates synthetic open-set pixels by pasting virtual objects (e.g., cropped from PASCAL VOC [51]) on Cityscapes images. In this work, we do not generate synthetic pixels but instead repurpose "other" pixels (outside the K closed-set classes) that already exist in Cityscapes. Interestingly, classic semantic segmentation benchmarks evaluate these "other" pixels as a separate background class [51], but Cityscapes ignores them in its evaluation (so do other contemporary datasets [56], [57], [58],

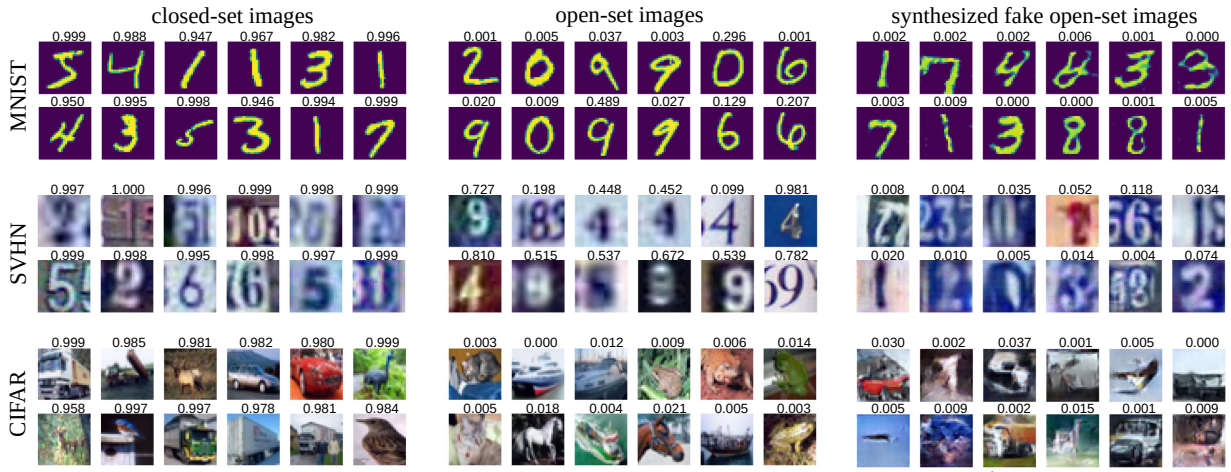


Fig. 3: Visual examples of closed-set, open-set, and fake open-set images synthesized by the OpenGAN-0^{pix} generator (for three datasets). We mark the classification confidence scores as probabilities of being recognized as the closed-set by the OpenGAN-0^{pix} discriminator. We can see the synthesized images look realistic in terms of color and shape. But the discriminator can easily recognize them as the open-set, as indicated by the low probabilities.

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. 4(f-i) for ROC curves, and F1-scores vs. thresholds on the open-set likelihood.

open-test	*metric*	MSP [20]	OpenMax [7]	NN^{fea} [41]	GMM [39]	C2AE [9]	MSP$_c$ [5]	MCdrop [26]	GDM [21]	CLS^{pix}	(K+1)	CLS^{fea}	Open GAN^{pix}	Open 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**

[59]). The historically-ignored pixels include vulnerable objects (e.g., strollers in Fig. 1), and can be naturally evaluated as open-set examples. We refer the reader to Appendix C for further details.

Dataset. Cityscapes [1] contains 1024x2048 high-resolution urban scene images with 19 classes 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. 1) are the open-set examples in this setup.

Pixel Generation. As Cityscapes images are high in resolution (1024x2048), it is nontrivial to train an OpenGAN^{pix} to generate high-res fake open-set 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. 7 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. Firstly, 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. 7). This suggests that OTS features (from the segmentation network) serve as a powerful representation for open-set pixel recognition. The curves in Fig. 7 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. Secondly, 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 C2AE in Table 5) which assume no open training data. This confirms the effectiveness of Outlier Exposure, even with a modest amount of outliers [4]. We refer the reader to Appendix D for hyperparameter tuning. Moreover, Bayesian networks (MCdrop and MSP$_c$) outperform the baseline MSP, showing that uncertainties can be reasonably used for open-set recognition. Lastly, we train

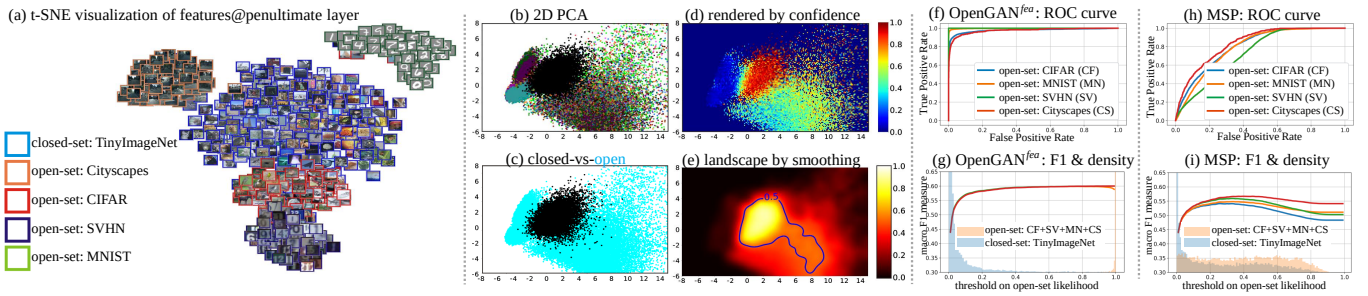


Fig. 4: **(a)** We use t-SNE to visualize the embedding space through the OTS features computed by the K-way network trained on TinyImageNet train-set. Images from the other datasets are open-set examples. Clearly, closed and open examples are well separated in the feature space. We further visualize 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).

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 [20], 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 outlier / open-set 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**

TABLE 5: **Comparison in open-set semantic segmentation** on Cityscapes (AUROC ↑). All methods are implemented on top of the segmentation network HRNet [2] except the ones operating on pixels (as marked by ^{pix}). Our approach OpenGAN^{fea} clearly performs the best. Fig. 7 analyzes OpenGAN trained with varied number of open-set pixels, when built on either pixels or OTS features.

MSP [20]	Entropy [52]	OpenMax [7]	C2AE [9]	MSP$_c$ [5]	MCdrop [26]	GDM [21]	GMM [39]	HRNet-(K+1)	OpenGAN-0^{fea}	CLS^{fea}	OpenGAN^{fea}
.721	.697	.751	.722	.755	.767	.743	.765	.755	.709	.861	**.885**

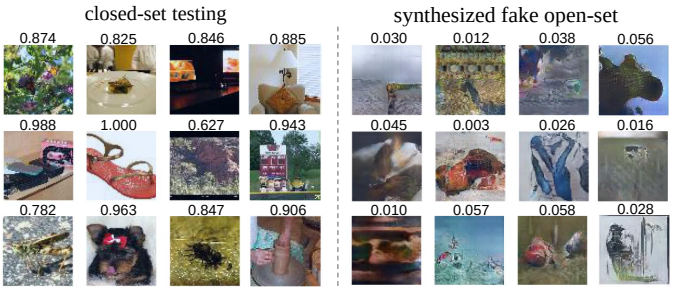


Fig. 5: **Visual examples of closed-set testing images and fake open-set images synthesized by OpenGAN-0^{pix}**. We mark the classification confidence scores above each image as the probability of being recognized as the closed-set. **Left**: The discriminator recognizes the closed-set training examples with a high confidence score. **Right**: OpenGAN-0^{pix} synthesizes fake images that look realistic in terms of color and tone but not content (i.e., they do not contain meaningful objects); the discriminator easily recognizes these fake images (as indicated by the low probability).

a "ground-up" (K+1)-way HRNet model that treats "other" pixels as the $(K+1)^{th}$ background class [51], shown by HRNet-(K+1) in Table 5. It performs better than other typical open-set 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. 6 qualitatively compares OpenGAN and the entropy method (more visual results are in the appendix). The visualization shows OpenGAN sufficiently recognize open-set pixels. It also implies failure happens when OpenGAN misclassifies open-vs-closed pixels. Fig. 8 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.

4.5 A Study of Model Selection

It is crucial and challenging to select a good discriminator for open-set discrimination due to the unstable training of GANs [36]. For GANs that are typically used for generating realistic images, one usually focuses on selecting generators by manually inspecting the generated visual images by different model epochs [17]. In contrast, we must select the discriminator, rather than the generator, because we use the discriminator as the open-set like-

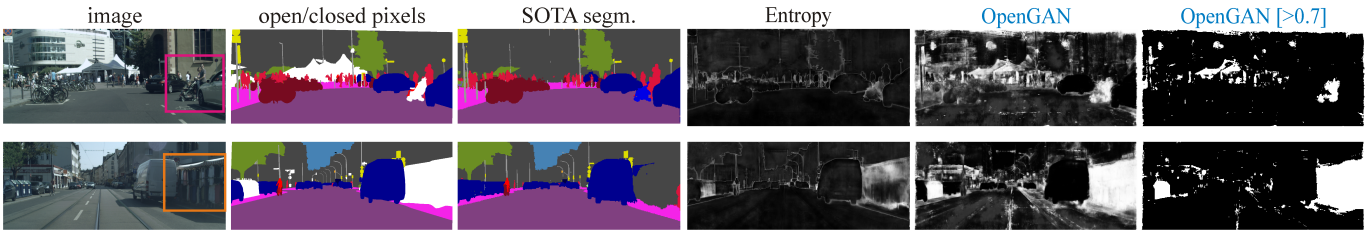


Fig. 6: Qualitative results of two testing images, on which a state-of-the-art network (HRNet) misclassifies the *unknown* categories stroller/street-shop as motorcycle/building. From left to right of each row: the input image, its per-pixel semantic labels (in which `white` regions are open-set pixels), 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).

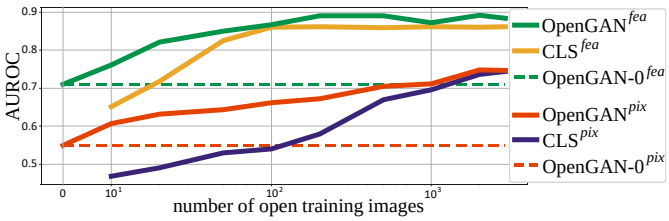


Fig. 7: Diagnostic study w.r.t AUROC vs. number of open images which provide open-set training pixels. Our methods perform better on OTS features than pixels. Recall OpenGAN-0 is equivalent to training a normal GAN (without open training data) and using its discriminator as open-set likelihood. With some open training data (e.g., 100 open images), CLS outperforms OpenGAN-0; but OpenGAN consistently performs the best.

Fig. 8: 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 patch for a generated feature by finding the closest pixel feature on the training set and returning its surrounding image 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.

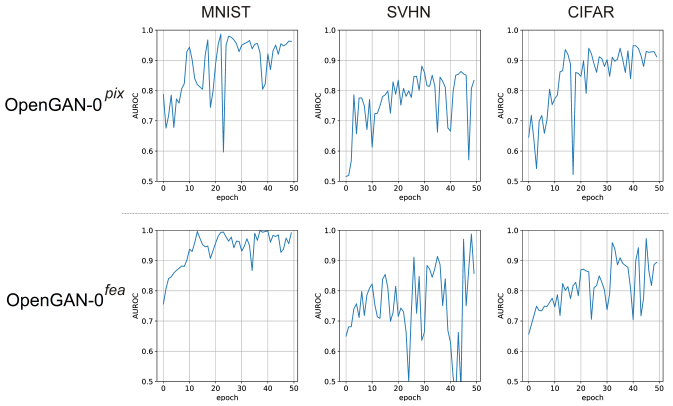


Fig. 9: **Open-set discrimination performance vs. training epochs**. We show the performance (AUROC) by OpenGAN-0^{pix} and OpenGAN-0^{fea} on the val-sets of the three datasets (cf. columns) under *Setup-I*. Recall that OpenGAN-0 trains a normal GAN and uses its discriminator as the open-set likelihood function. We can see that many intermediate checkpoints of GAN-discriminators achieve very high open-set discrimination accuracy, but longer training does *not* necessarily improve further. This is due to the unstable training of GANs via the minimax optimization. This motivates model selection for the discriminators.

lihood function. It is important to note that, when reaching the equilibrium of the minimax optimization between the discriminator and generator in GAN training, the discriminator would be incapable of recognizing fake open-set data [60]. Fortunately, we find that model selection using a validation set can select an intermediate discriminator that produces the state-of-the-art open-set classification performance. Below we present the need of model selection and demonstrate using synthetic data is inadequate for model selection.

Model selection is crucial. In Fig. 9, we plot the open-set classification performance as a function of training epochs. We study both OpenGAN-0^{fea} and OpenGAN-0^{pix} on the three datasets as typically used in open-set recognition (under *Setup-I*). Recall that OpenGAN-0 trains a normal GAN and uses its discriminator as the open-set likelihood function for open-set discrimination. Clearly, longer training time does *not* necessarily improve open-set classification performance. We posit that this is due to the unstable training of GANs. This motivates the model selection for discriminators.

Synthesized data are not sufficient for model selection. To study how each checkpoint models perform in training (fake-vs-real classification) and testing (open-vs-closed classification), we scatter-plot Fig. 10, where we render the dots with colors to indicate the model epoch (blue→red dots represent model epoch-0→50). For the scatter plots, the ideal case is that the train-time and test-time accuracy is correlated, i.e., all dots appear in the diagonal line (from origin to top right). But the plots demonstrate the opposite, suggesting that using the synthesized data for model selection is not sufficient. Instead, we find it crucial to use a validation set of real open examples to select the open-set discriminator. Our observation is consistent to what reported in [4]. It is worth noting that the models selected on the validation set *do* generalize to test sets, as demonstrated by Table 3 and 4.

4.6 Failure Cases and Limitations

As we use a discriminator as the open-set likelihood function, failure cases happen when the open-vs-closed classification is not correct, as shown by some high confidence scores for open-set images in Fig. 3.

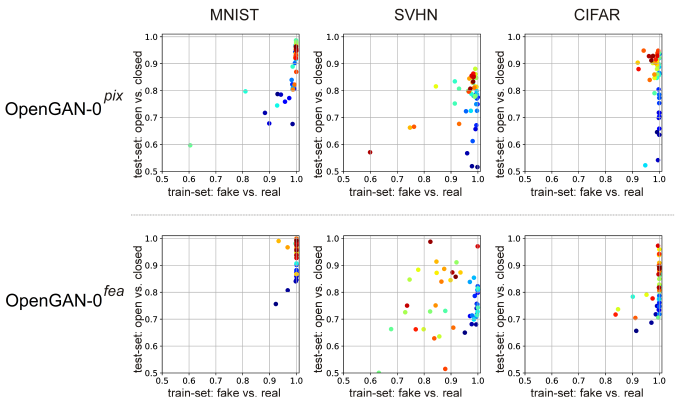


Fig. 10: **Scatter plot of training performance (fake-vs-real classification accuracy in AUROC, x-axis) and testing performance (open-vs-closed classification accuracy in AUROC, y-axis).** We train OpenGAN-0 models on the three datasets (cf. columns) under *Setup-I*. A dot corresponds to a particular model checkpoint during training. We color the dots from blue→red to mark models saved at epoch-0→50. The ideal correlation between training and testing accuracy is that all the dots lie in the diagonal from bottom-left to top-right, and blue / red dots are on the bottom-left / top-right. Clearly, they do not correlate well. Moreover, the dots appear to be on the right part in the plots, suggesting that the training performance is much higher than the testing performance. These scatter plots demonstrate that (1) some intermediate discriminators can perform quite well in open-set discrimination but (2) it requires careful selection of the right discriminator, and (3) fake data synthesized by the GAN-generators are insufficient to be used for model selection.

We point out other failure cases and limitations. First, although we empirically show superior performance by model selection on a validation set, there exist risks that the validation set is biased in some way which could catastrophically hurt the final open-set recognition performance. This is also true for the training outliers which will not span the open world. Therefore, in the real open-world, practitioners should be aware of such a bias and exploit prior knowledge in constructing "reliable" training and validation sets for training OpenGANs. Second, as we adopt adversarial training for OpenGANs, it is straightforward to ask if OpenGAN is robust to adversarial perturbations on the input images. We have not investigated this point and leave it as future work.

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.

Acknowledgments

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

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.
[2] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE PAMI*, 2019.
[3] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *ICLR*, 2018.
[4] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *ICLR*, 2019.
[5] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *ICLR*, 2018.
[6] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE PAMI*, 2012.
[7] A. Bendale and T. E. Boult, "Towards open set deep networks," in *CVPR*, 2016.
[8] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in *CVPR*, 2019.
[9] P. Oza and V. M. Patel, "C2AE: class conditioned auto-encoder for open-set recognition," in *CVPR*, 2019.
[10] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*, 2017.
[11] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *CVPR*, 2018.
[12] S. Pidhorskyi, R. Almohsen, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *NeurIPS*, 2018.
[13] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *IEEE International Conference on Data Mining (ICDM)*, 2018.
[14] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, 2019.
[15] A. R. Dhamija, M. Günther, and T. Boult, "Reducing network agnostophobia," in *NeurIPS*, 2018.
[16] A. Shafaei, M. Schmidt, and J. J. Little, "A less biased evaluation of out-of-distribution sample detectors," in *BMVC*, 2019.
[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.
[18] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, 2009.
[19] C. Geng, S.-j. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE PAMI*, 2020.
[20] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *ICLR*, 2017.
[21] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *NeurIPS*, 2018.
[22] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," in *BMVC*, 2017.
[23] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *ECCV*, 2018.
[24] X. Sun, Z. Yang, C. Zhang, K.-V. Ling, and G. Peng, "Conditional gaussian distribution learning for open set recognition," in *CVPR*, 2020.
[25] H. Zhang, A. Li, J. Guo, and Y. Guo, "Hybrid models for open set recognition," in *ECCV*, 2020.
[26] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016.
[27] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *ICCV*, 2019.
[28] D. Dehaene, O. Frigo, S. Combrexelle, and P. Eline, "Iterative energy-based projection on a normal data manifold for anomaly localization," in *ICLR*, 2020.
[29] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *ICCV*, 2015.
[30] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *ACCV*, 2018.
[31] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, "Image anomaly detection with generative adversarial networks," in *ECML/PKDD*, 2018.

[32] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," in *ICLR*, 2020.
[33] A. Bendale and T. Boult, "Towards open world recognition," in *CVPR*, 2015.
[34] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" in *ICLR*, 2018.
[35] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, and Y. Tian, "Learning open set network with discriminative reciprocal points," in *ECCV*, 2020.
[36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*, 2017.
[37] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *ICLR*, 2019.
[38] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *ICLR*, 2018.
[39] S. Kong and D. Ramanan, "An empirical exploration of open-set recognition via lightweight statistical pipelines," 2021. [Online]. Available: https://openreview.net/forum?id=0Zxk3ynq7jE
[40] P. Perera, V. I. Morariu, R. Jain, V. Manjunatha, C. Wigington, V. Ordonez, and V. M. Patel, "Generative-discriminative feature representations for open-set recognition," in *CVPR*, 2020.
[41] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *SIGMO*, 2000.
[42] P. Perera and V. M. Patel, "Deep transfer learning for multiple class novelty detection," in *CVPR*, 2019.
[43] A. Torralba and A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011.
[44] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, "The fishyscapes benchmark: Measuring blind spots in semantic segmentation," *arXiv:1904.03215*, 2019.
[45] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. Song, "A benchmark for anomaly segmentation," *arXiv:1911.11132*, 2019.
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[47] J.-Y. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.
[48] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.
[49] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," *NIPS Workshop Autodiff*, 2017.
[50] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *ICML*, 2006.
[51] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *IJCV*, 2015.
[52] J. Steinhardt and P. S. Liang, "Unsupervised risk estimation using only conditional independence structure," in *NeurIPS*, 2016.
[53] W. J. Scheirer, A. Rocha, R. J. Micheals, and T. E. Boult, "Meta-recognition: The theory and practice of recognition score analysis," *IEEE PAMI*, 2011.
[54] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Robotics and Automation Letters*, 2020.
[55] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient gan-based anomaly detection," *arXiv:1802.06222*, 2018.
[56] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.
[57] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *ICCV*, 2019.
[58] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, 2016.
[59] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *ICCV*, 2017.
[60] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," in *ICML*, 2017.
[61] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, 2008.
[62] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *ICLR*, 2019.
[63] S. Yang and D. Ramanan, "Multi-scale recognition with dag-cnns," in *CVPR*, 2015.
[64] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *ECCV*, 2014.
[65] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *IJCV*, 2017.
[66] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *ECCV*, 2012.
[67] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *CVPR*, 2008.
[68] P. R. M. Júnior, R. M. De Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha, "Nearest neighbors distance ratio open-set classifier," *Machine Learning*, 2017.
[69] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, "Deep quantization network for efficient image retrieval," in *AAAI*, 2016.
[70] S. Alshammari, Y. Wang, D. Ramanan, and S. Kong, "Long-tailed recognition via weight balancing," in *CVPR*, 2022.
[71] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, 2018.

APPENDIX A
STATISTICAL MODELS FOR OPEN-SET

In this section, we introduce a lightweight statistical pipeline that repurposes off-the-shef (OTS) deep features for open-set recognition. While details can be found in [39], the pipeline briefly follows: (1) extracting OTS features (with appropriate processing described below) of closed-set training examples using the underlying K-way classification model, (2) learning statistical models over the OTS features. There are many statistical methods readily available, such as nearest class centroids, Nearest Neighbors, and (class-conditional) Gaussian Mixture Models (GMMs). During testing, we extract the OTS features of the given example and use the learned statistical models to compute an open-set likelihood, e.g., based on the (inverse) closed-set probability from GMM. By thresholding the open-set likelihood, we decide whether it is an open-set example.

Feature extraction. OTS features generated at different layers of the trained K-way classification network can be repurposed for open-set recognition. Most methods leverage softmax [20] and logits [7], [9], [62] which can be thought of as features extracted at top layers. Similar to [21], we find it crucial to analyze features from intermediate layers for open-set recognition, because logits and softmax may be too invariant to be effective for open-set recognition (Fig. 11). One immediate challenge to extract features from an intermediate layer is their high dimensionality, e.g. of size 512x7x7 from ResNet18 [46]. To reduce feature dimension, we simply (max or average) pool the feature activations spatially into a 512-dim feature vectors [63]. We can further reduce dimension by applying PCA, which can reduce dimensionality by $10\times$ (from 512-dimensional to 50 dimensional) without sacrificing performance. We find this dimensionality particularly important for learning second-order covariance statistics as in GMM, described below. Finally, following [64], [65], we find it crucial to L2-normalize extracted features (Fig. 11). We refer the reader to [39] for quantitative results.

Statistical models. Given the above extracted features, we use various generative statistical methods to learn the confidence/probability that a test example belongs to the closed-set

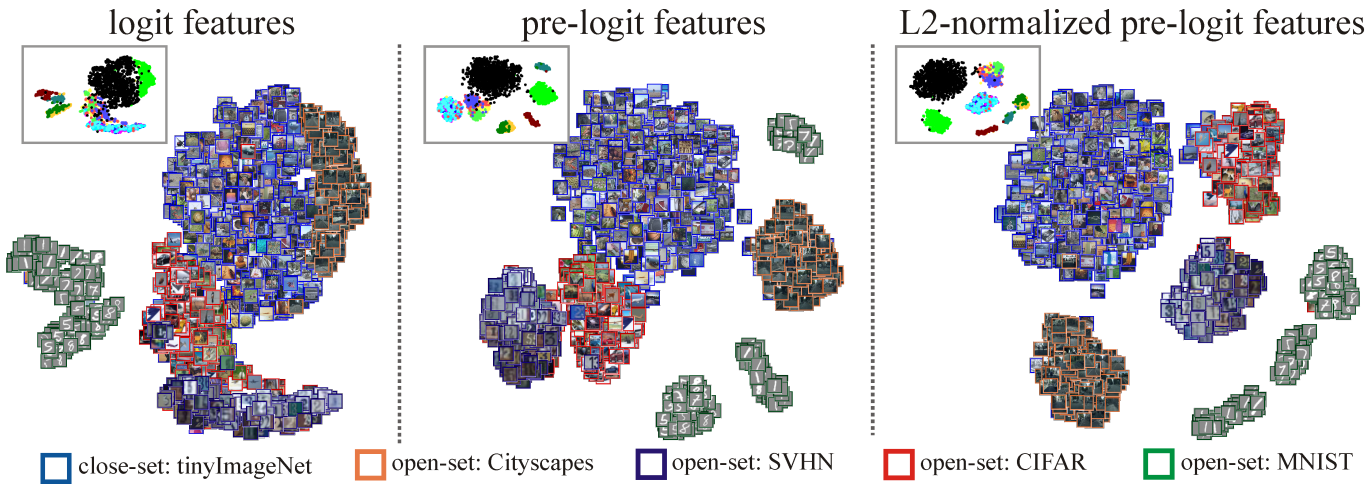


Fig. 11: t-SNE plots [61] of open vs. closed-set testing data, as encoded by different features from a ResNet18 network trained from scratch on the 200-way TinyImageNet dataset (under *Setup-II*). In the top-left subfigures of each of the three panels, we plot black dots as the closed-set examples from TinyImageNet, and randomly color the open-set examples using their class labels provided by the respective datasets. **Left**: Logit features mix open- and closed-set data, suggesting that methods based on them (e.g., Entropy, SoftMax and OpenMax) may struggle in open-set discrimination. **Mid**: Pre-logit features at the penultimate layer show better separation between closed- and open-set data. **Right**: L2-normalizing the pre-logits features separates them better. These plots intuitively demonstrate the benefit of L2-normalization and using OTS features rather than the highly-invariant logits for open-set discrimination.

classes. Such statistical methods include simple parametric models such as class centroids [66] and class-conditional Gaussian models [21], [62], non-parametric models such as NN [67], [68], and mixture models such as (class-conditional) GMMs and k-means [69]. A statistical model labels a test example as open-set when the inverse probability (e.g., of the most-likely class-conditional GMM) or distance (e.g., to the closest class centroid) is above a threshold. One benefit of such simple statistical models is that they are interpretable and relatively easier to diagnose failures. For example, one failure mode is an open-set sample being misclassified as a closed-set class. This happens when open-set data lie close to a class-centroid or Gaussian component mean (see Fig. 11). Note that a single statistical model may have several hyperparameters – GMM can have multiple Gaussian components and different structures of second-order covariance, e.g., either a single scalar, a diagonal matrix or a general covariance per component. We use the validation set to set these hyperparameters, as opposed to prior works that conduct model selection either unrealistically on the test-set [9] or on large-scale val-set which could be arguably used for training [21].

Lightweight Pipeline. We emphasize that the above feature extraction and statistical models result in a lightweight pipeline for open-set recognition. To understand this, we analyze the number of parameters involved in the pipeline. Assume we learn a GMM over 512x7x7 feature activations, and specify a general covariance and five Gaussian components. If we learn the GMM directly on the feature activations, the number of parameters from the second-order covariance alone is at the scale of $(512 * 7 * 7)^2$. With the help of our feature extraction (including spatial pooling and PCA), we have 50-dim feature vectors, and the number of parameters in the covariance matrices is now at the scale of 50^2. This means a huge reduction (10^5 $\times$) in space usage! We count the total number of parameters in this GMM: 3.3$\times$ 10^4 32-bit float parameters including PCA and GMM's five components, amounting to 128KB storage space. Moreover, given that PCA just runs once for all classes, even when we learn such GMMs for each of 19 classes (as those defined in Cityscapes), it only requires 594KB storage space! Compared to the modern networks such as HRNet (>250MB), our statistical pipeline for open-set recognition adds a negligible (0.2%) amount of compute, making it quite practical for implementation on autonomy stacks.

APPENDIX B
MODEL ARCHITECTURE

We describe the network architectures of OpenGAN. Recall that our final version OpenGAN^{fea} operates on off-the-shelf (OTS) features. We use multi-layer perceptron (MLP) networks for the generator and discriminator. As OpenGAN^{pix} operates on pixels, we adopt convolutional neural network (CNN) architectures. We begin with the former.

B.1 OpenGAN^{fea} architecture

OpenGAN^{fea} consists of a generator and a discriminator. OpenGAN^{fea} is compact in terms of model size (∼2MB), because it learns MLPs over OTS features which are low-dimensional (e.g., 512-dim vectors). The MLP architectures are described below.

- The MLP discriminator in OpenGAN^{fea} takes a D-dimensional feature as the input. Its architecture has a set of fully-connected layers (fc marked with input and output dimensions), Batch Normalization layers (BN) and LeakyReLU layers (hyper-parameter as 0.2): `fc (D→64*8), BN, LeakyReLU, fc (64*8→64*4), BN, LeakyReLU, fc (64*4→64*2), BN, LeakyReLU, fc (64*2→64*1), BN, LeakyReLU, fc (64*1→1), Sigmoid.`
- The MLP generator synthesizes a D-dimensional feature given a 64-dimensional random vector: `fc (64→64*8), BN, LeakyReLU, fc (64*8→64*4), BN, LeakyReLU, fc (64*4→64*2), BN, LeakyReLU, fc (64*2→64*4), BN, LeakyReLU, fc (64*4→D), Tanh.`

For open-set image classification, the image features have dimension $D = 512$ from ResNet18 (the K-way classification

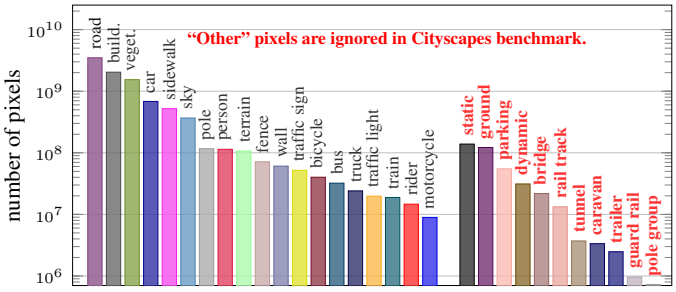


Fig. 12: Cityscapes annotates a sizeable portion of pixels that do not belong to the K closed-set classes on which the Cityscapes benchmark evaluates. As a result, many methods ignore them during training [2]. We repurpose these historically-ignored pixels as open-set examples that are from the $(K{+}1)^{th}$ "other" class, allowing for a large-scale exploration of open-set recognition via semantic segmentation.

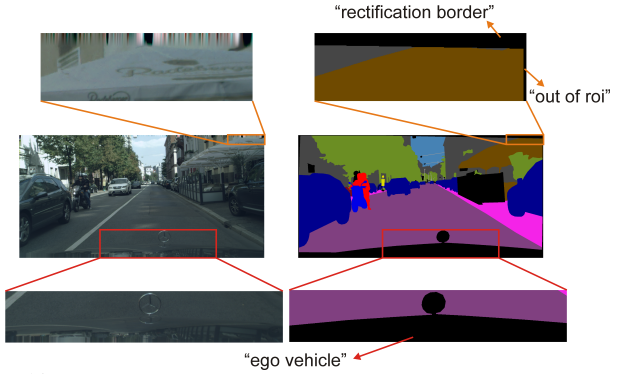


Fig. 13: **Void pixels in Cityscapes** that are not from the closed-set classes nor open-set. We highlight these pixels over an image (left) and its semantic annotations (right). Cityscapes labels these pixels as `rectification-border` (artifacts at the image borders caused by stereo rectification), `ego-vehicle` (a part of the car body at the bottom of the image including car logo and hood) and `out-of-roi` (narrow strip of 5 pixels along the image borders). As these void pixels can be easily filtered out, we do not evaluate on them.

networks under *Setup-I* and *II*). For open-set semantic segmentation (*Setup-III*), the per-pixel features have dimension $D = 720$ at the penultimate layer of HRnet [2].

B.2 OpenGAN^{pix} architecture

OpenGAN^{pix}'s generator and discriminator follow the CycleGAN architecture [47]. We change the stride size in the convolution layers to adapt the networks to specific image resolution (e.g., CIFAR 32x32 and TinyImageNet 64x64). The generator and discriminator in OpenGAN^{pix} have model sizes as ∼14MB and ∼11MB, respectively. We find it important to ensure that OpenGAN^{pix} has a larger capacity than OpenGAN^{fea} to generate high-dimensional RGB raw images.

APPENDIX C
SETUP FOR OPEN-SET SEMANTIC SEGMENTATION

We use Cityscapes to study open-set semantic segmentation. Prior work suggests pasting virtual objects (e.g., cropped from PASCAL VOC masks [51]) on Cityscapes images as open-set pixels [44], [45]. We notice that Cityscapes ignores a sizeable portion of pixels in its benchmark (presumably because of rare-classes have too few examples to reliably learn their classifiers [70], [71], as demonstrated by Fig. 12). As a result, many methods also ignore

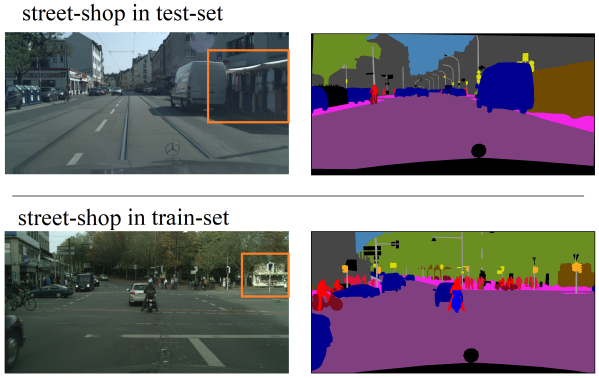


Fig. 14: **street-shop as open-set** (left: RGB image; right: pixel annotations). Fig. 1 shows open-set pixel recognition results for a street-shop on a testing image (top-row). We verify that such a street-shop does not appear in the training set. After manual inspection of the training set, we find the one (bottom-row) most similar to the testing example in terms of size. Importantly, we did not find any other street-shops in the training set that sell clothes like the one in the top row. In this sense, this street-shop is a real open-set example.

them in training. Therefore, instead of introducing artificial open-set examples, we use the historically-ignored pixels in Cityscapes as the real open-set examples. One might argue that testing open-set examples are seen during training and hence not strictly open-set anymore. But we argue that this is a realistic way to leverage the real "known unknowns" (via the "other" class) for open-set recognition. We further manually inspect recognized open-set pixels and find that the detected open-set regions such as the "street-shop" in Fig. 14 are real open-set that no training examples are like such. We describe in detail our configuration for open-set semantic segmentation setup and experiments on Cityscapes.

Data Setup. Cityscapes training set has 2,975 images. We use the first 2,965 images for training and the last 10 as valset for model selection. We use the 500 Cityscapes validation images as our test set. Below are the statistics for the full train/val/test sets.

- train-set for closed-pixels: 2,965 images providing 334M closed-set pixels.
- train-set for open-pixels: 2,965 images providing 44M open-set pixels.
- val-set for closed-pixels: 10 images providing 1M closed-set pixels.
- val-set for open-pixels: 10 images providing 0.2M open-set pixels.
- test-set for closed-pixels: 500 images providing 56M pixels.
- test-set for open-pixels: 500 images providing 2M pixels.

Note that, we exclude the pixels labeled with `rectification-border` (artifacts at the image borders caused by stereo rectification), `ego-vehicle` (a part of the car body at the bottom of the image including car logo and hood) and `out-of-roi` (narrow strip of 5 pixels along the image borders). These pixels can be easily localized using camera information, as demonstrate in Fig. 13. We do not evaluate on such pixels.

Feature setup.

- M^{pix}, where $M \in \{\text{CLS, OpenGAN}\}$, corresponds to a model defined on raw pixels.
- M^{fea} corresponds to a model defined on embedding features at the penultimate layer of underlying semantic segmentation network (i.e., HRNet as introduced below).
- HRNet [2] is a top-ranked semantic segmentation model on Cityscapes. It has a multiscale pyramid head that produce

TABLE 6: **Hyper-parameter tuning for open-set semantic segmentation on Cityscapes.** Given a fixed number of open training images, we vary the hyper-parameter λ_G to train OpenGAN models. Recall that λ_G controls the contribution of synthesized data in the loss function. We conduct model selection on the val-set (10 images), and report here the performance (AUROC↑) on the test set (500 images). We also mark the λ_G for each of the selected models. It seems to be preferable to set a lower weight λ_G (for the term exploiting synthesized data examples) when we have more real open-set data, but we do not see a tight correlation between λ_G and test-time performance. We believe this is because of the (random) initialization of model weights that has a non-trivial impact on training GANs and their final performance.

$\#\text{images}^{open}_{train}$	10	20	50	100	200	500	1000	2000	2900
OpenGAN^{fea}	.761	.821	.849	.866	.891	.890	.873	.891	.885
λ_G	0.20	0.20	0.05	0.10	0.05	0.05	0.20	0.05	0.05
OpenGAN^{pix}	.607	.632	.643	.661	.672	.705	.711	.748	.746
λ_G	0.60	0.40	0.80	0.90	0.70	0.60	0.60	0.60	0.70

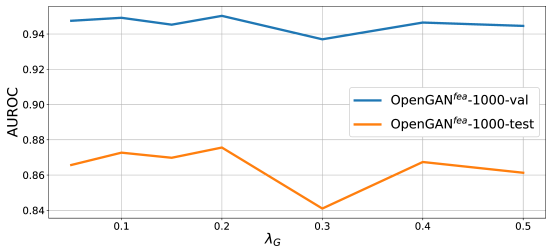


Fig. 15: **Tuning hyper-parameter** λ_G. We plot the open-set discrimination performance (AUROC) as a function of λ_G, which controls the contribution of generated data in the loss function. We report the OpenGAN^{fea}-1000 model which is trained with 1000 training images. The val-set and test-set contain 10 and 500 images respectively. While the val-set is much smaller than the test-set, its open-set classification accuracy aligns well with that on the test set.

high-resolution segmentation prediction. We extract embedding features at its penultimate layer (720-dimensional before the 19-way classifier). We also tried other layers but we did not observe significant difference in their performance.

Batch Construction. To fully shuffle open- and closed-set training pixels, we cache all the open-set training pixel features extracted from HRNet. We construct a batch consisting of 10,000 pixels for training OpenGAN^{fea}. To do so, we

- randomly sample a real image, run HRNet over it and randomly extract 5,000 closed-set training pixel features;
- randomly sample 2,500 open-set training features from cache;
- run the OpenGAN^{fea} generator (being trained on-the-fly) to synthesize 2,500 "fake" open-set pixel features.

Similarly, to train OpenGAN^{pix} which is fully-convolutional, we construct a batch of 10,000 pixels as below.

- We feed a random real image to the OpenGAN^{pix} discriminator, and penalize predictions on 5000 random closed pixels and 2500 random open pixels.
- We run the OpenGAN^{pix} generator (being trained on-the-fly) to synthesize a "fake" image. We feed this "fake" image to the discriminator along with *open-set* labels. We penalize 2500 random "fake" pixels.

APPENDIX D
HYPER-PARAMETER TUNING

Strictly following the practice of machine learning, we tune hyper-parameters on the validation set. We study parameter tuning through open-set semantic segmentation (*Setup-III*). We select the best OpenGAN model according to the performance on the valset (10 images).

In training OpenGAN, a training batch contains both real closed- and open-set pixels, and synthesized *fake open* pixels. Correspondingly, our loss function has three terms (cf. Eq. 2). Therefore, we tune the hyper-parameter λ_o and λ_G as below to balance the terms in the loss function that exploits real open data and generated data:

- The term exploiting real open data has a weight $\lambda_o = 1$. We do not tune this as we presume the sparsely sampled open-set examples are equally important as the real closed-set examples.
- The term using the generated "fake" data has varied parameter $\lambda_G \in [0.05, 0.10, 0.15, \ldots, 0.85, 0.90]$. We mainly focus on tuning λ_G to study how the synthesized data help training.

Table 6 shows the performance of OpenGAN^{pix} and OpenGAN^{fea} on the test-set with varied training images. For each selected model, we mark the corresponding λ_G that yields the best performance (on the valset). Roughly speaking, it is preferable to set a lower weight λ_G when we have more real open-set training data. However, we do not see a clear correlation between the weight λ_G and test-time performance. We believe this is due to the random initialization which affects adversarial learning.

We also study how models trained with different λ_G perform on the val-set and test-set, and if the model selected on the val-set can reliably perform well on the test-set. Fig. 15 plots the performance as a function of λ_G on val-set and test-set. We use the OpenGAN^{fea} model trained with 1000 training images (containing closed- and open-set pixels). We can see the performance on the validation set reliably reflects the performance on the test set. This confirms that model selection on the val-set is reliable.