# *OpenGAN*: Open-Set Recognition via Open Data Generation
## *(Supplementary Material)*

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

## Outline

*As elaborated in the main paper, our proposed OpenGAN trains an open-vs-closed binary classifier for open-set recognition. Our three major technical insights are (1) model selection of a GAN-discriminator as the open-set likelihood function via validation, (2) augment the available set of real open training examples with adversarially synthesized "fake" data, and (3) training OpenGAN on off-the-shelf (OTS) features rather than pixel images. We expand on the techniques of OpenGAN in the appendix, including architecture design, model selection and additional details for training. We also provide additional comparisons to recently published methods and qualitative results. Below is the outline.*

**Section 1: Model architectures** for both OpenGAN$^{fea}$ and OpenGAN$^{pix}$.

**Section 2: Detailed setup for open-set semantic segmentation**, such as data statistics (e.g., the number of open-set pixels in the testing set) and batch construction during training.

**Section 3: Model selection** that is performed on a validation set.

**Section 4: Hyper-parameter tuning** which is performed on a validation set.

**Section 5: Statistical methods for open-set recognition** that learn generative models (e.g., Gaussian Mixture) over off-the-shelf deep features.

**Section 6: More quantitative comparisons** to several approaches published recently.

**Section 7: Visuals of synthesized images** generated by OpenGAN-0$^{pix}$ and OpenGAN-0$^{fea}$, intuitively demonstrating their effectiveness and limitations.

**Section 8: Visual results of open-set semantic segmentation**.

**Section 9: Failure Cases and Limitations**.

## 1. Model Architecture

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

### 1.1. OpenGAN$^{fea}$ architecture

OpenGAN$^{fea}$ consists of a generator and a discriminator. OpenGAN$^{fea}$ is compact in terms of model size ($\sim$2MB), because it adopts MLP network over OTS features which are low-dimensional (e.g., 512-dim vectors) compared to pixel images. 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-dimension and output-dimension), 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 networks under *Setup-I* and *II*). For open-set segmentation, the per-pixel features have dimension $D = 720$

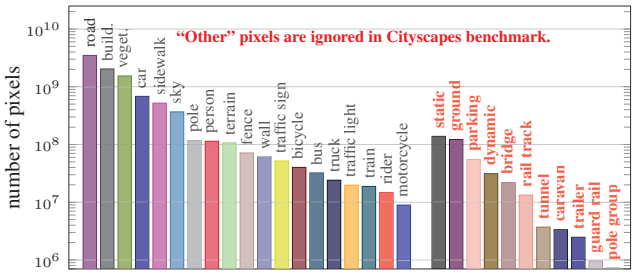


Figure 1: Cityscapes annotates a sizeable portion of pixels that do not belong to one of the $K$ closed-set classes on which the Cityscapes benchmark evaluates. As a result, many methods also ignore them during training [29]. We repurpose these historically-ignored pixels as open-set examples that are from the $(K{+}1)^{th}$ "other" class, allowing for a large-scale exploration of open-set recognition via semantic segmentation.

at the penultimate layer of HRnet (a top-ranked semantic segmentation model used in this work under *Setup-III*).

### 1.2. OpenGAN$^{pix}$ architecture

OpenGAN$^{pix}$'s generator and discriminator follow the CycleGAN architecture[32]. 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 $\sim$14MB and $\sim$11MB, respectively. We find it important to ensure that OpenGAN$^{pix}$ has a larger capacity than OpenGAN$^{fea}$ to generate high-dimensional RGB raw images.

## 2. Setup for Open-Set Semantic Segmentation

In this work, we use Cityscapes to study open-set semantic segmentation. Prior work suggests pasting virtual objects (e.g., cropped from PASCAL VOC masks [8]) on Cityscapes images as open-set pixels [4, 16]. We notice that Cityscapes ignores a sizeable portion of pixels in its benchmark, as demonstrated by Figure 1. As a result, many methods also ignore 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. We hereby 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 hold out the last 10 as validation set for model selection. We use the 500 Cityscapes validation images as our test set. Here are the statistics for the full train/val/test sets.

- train-set for closed-pixels: 2,965 images providing 334M closed-set pixels.

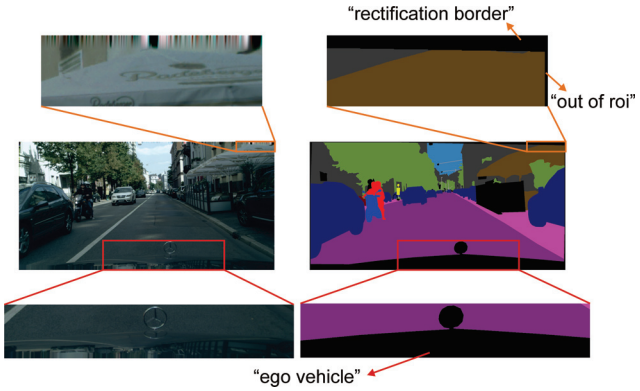


Figure 2: **Void pixels in Cityscapes** that are not from the closed-set classes nor open-set. We highlight these pixels over an image (left) and its semantic annotations (right). Cityscapes labels these pixels as `rectification-border` (artifacts at the image borders caused by stereo rectification), `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). These noise-pixels can be easily identified without machine-learned methods. Therefore, we do not evaluate on these 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. We demonstrate such pixels in Figure 2. In this sense, these pixels are not *unknown* open-set pixels but *known* noises caused by sensors and viewpoint. Therefore, we do not include them for open-set evaluation.

**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 seman-

tic segmentation network (i.e., HRNet as introduced below).

- HRNet [29] is a top-ranked semantic segmentation model on Cityscapes. It has a multiscale pyramid head that produce 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.

## 3. Model Selection

Due to the unstable training of GANs [1], model selection is crucial and challenging. GANs are typically used for generating realistic images, so model selection for GANs focuses on selecting generators. To do so, one relies heavily on manual inspection of visual results over the generated images from different model epochs [12]. In contrast, we must select the discriminator, rather than the generator, because we use the discriminator as an open-set likelihood function for open-set recognition. It is important to note that, in theory, a perfectly trained discriminator would not be capable of recognizing fake open-set data because of the equilibrium in the discriminator/generator game [2]. Although such an equilibrium hardly exist in practice, we find it crucial to select GAN discriminators to be used as open-set likelihood function. For model selection, we further find it crucial to use a validation set that consists of both real open and closed data. We present this study below.

**Model selection is crucial.** In Figure 3, 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 is to train a normal GAN and use its discriminator as open-set likelihood function for open-set recognition. Clearly, we can see that long training time does *not* necessarily improve open-set classification performance. We posit that this is due to the unstable training of GANs. This motivates robust model selection using a validation set.

**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 Figure 4, where we render the dots with colors to indicate the model epoch (blue$\rightarrow$ red dots represent model epoch-0$\rightarrow$50, respectively). For the scatter plot, the ideal case is that the train-time and test-time performance is linearly correlated, i.e., all dots appear in the diagonal line (from origin to top right). But their performances on the two sets are not correlated, 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 [18]. It is worth noting that the models selected on the validation set *do* generalize to test sets. This has been demonstrated in Table 3 and 4 in the main paper.

## 4. Hyper-Parameter Tuning

Strictly following the practice of machine learning, we tune hyper-parameters on the same validation set. We now study parameter tuning through open-set semantic segmentation (*Setup-III*). We select the best OpenGAN model according to the performance on the validation set (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 (refer to Eq. 2 in the main paper). Therefore, we tune the hyper-parameter $\lambda_o$ and $\lambda_G$ 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

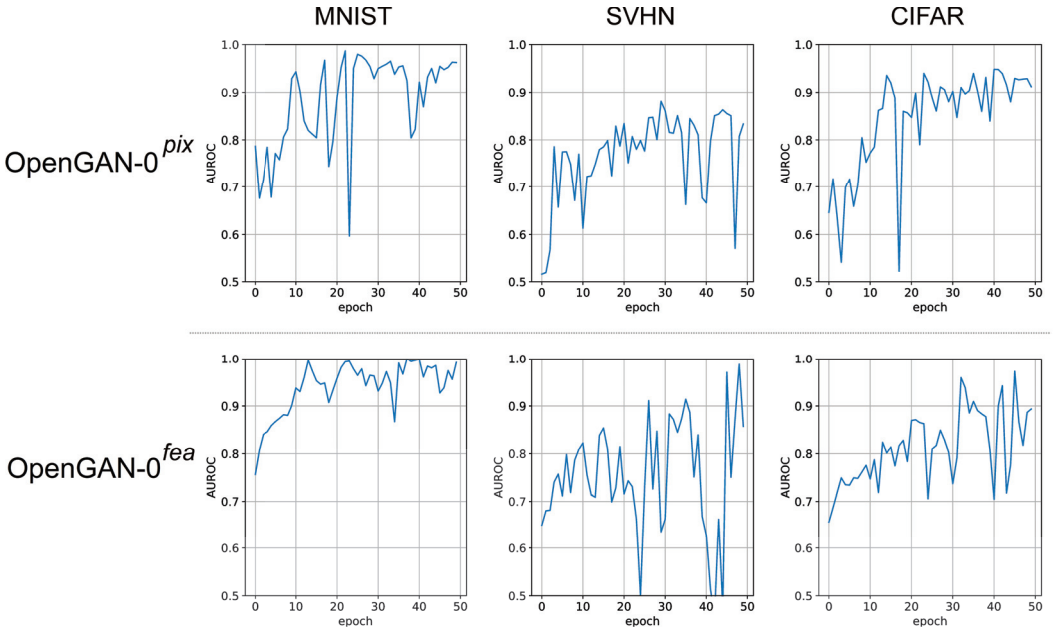


Figure 3: **Open-set image recognition performance vs. training epochs**. We show the performance (AUROC) by OpenGAN-$0^{pix}$ and OpenGAN-$0^{fea}$ on the val-sets of the three datasets which are widely studied in the open-set recognition literature (*Setup-I*). Recall that OpenGAN-0 is to train a normal GAN and use its discriminator as open-set likelihood function for open-set recognition. We can see that best open-set discrimination performance is achieved by intermediate checkpoints of GAN discriminators, and longer training does *not* necessarily improve performance. This is due to the unstable training of GANs with the min-max game. This motivates the need for robust model selection.

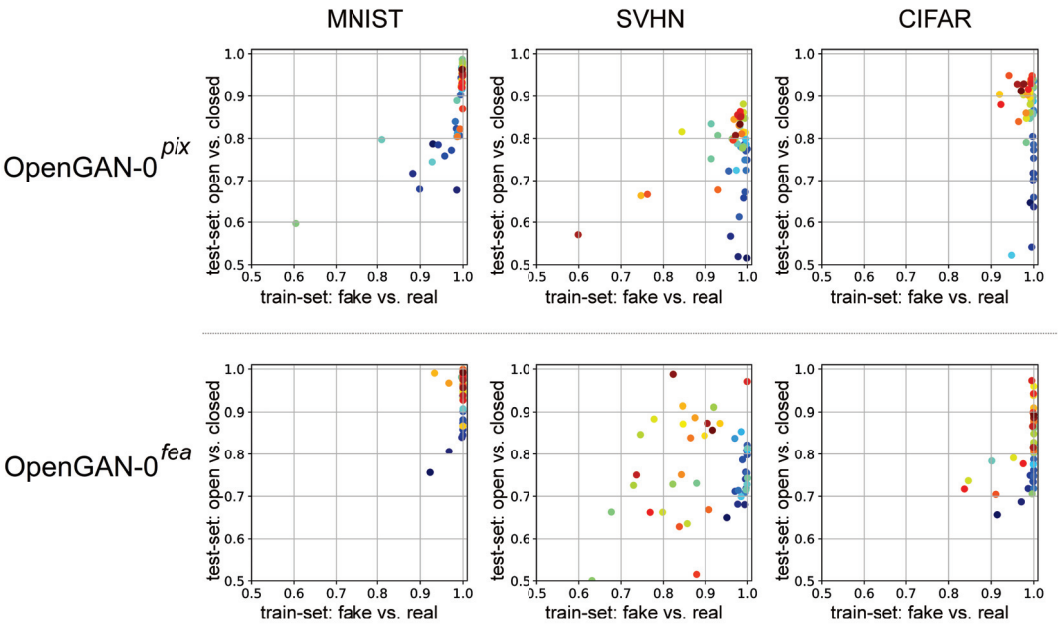


Figure 4: **Scatter plot of training performance (fake-vs-real classification) and testing performance (open-vs-closed classification).** We color the dots from blue→ red to marks models saved at epoch-0→50, respectively. We use the three datasets (under *Setup-I*) following the typical setup of open-set recognition. The ideal correlation is that all the dots lie in the diagonal from bottom-left to top-right. However, there is no correlation between training (fake-vs-real) and validation (open-vs-closed) performance. Moreover, because the dots appear to be on the right part in the plots, this means that fake-vs-real classification (as denoted by the $x$-axis) is much easier than open-vs-closed classification (as denoted by the $y$-axis). These scatter plots demonstrate that (1) intermediate discriminators can perform quite well in open-set discrimination (i.e., on the validation set consisting of real open and closed-set images), and (2) synthesized data are insufficient to be used for model selection.

Table 1: **Hyper-parameter tuning for open-set semantic segmentation on Cityscapes.** Given a fixed number of open training images, we vary the hyper-parameter $\lambda_G$ to train OpenGAN models. Recall that $\lambda_G$ 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 $\lambda_G$ for each of the selected models. It seems to be preferable to set a lower weight $\lambda_G$ (for the term exploiting synthesized data examples) when we have more real open-set data, but we do not see a tight correlation between $\lambda_G$ 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.

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

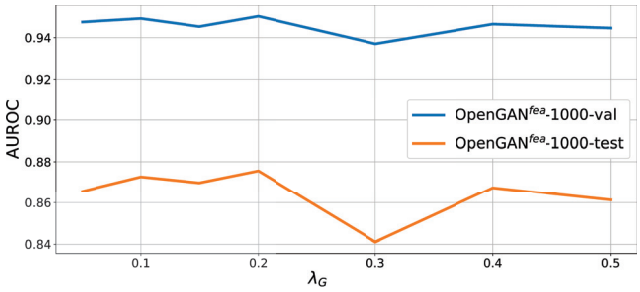


Figure 5: **Tuning hyper-parameter $\lambda_G$.** We plot the open-set discrimination performance (AUROC) as a function of $\lambda_G$, which controls the contribution of generated data examples in the loss function. The model we report here is $\text{OpenGAN}^{fea}$-1000 that is trained with 1000 open-set training images. The validation set and test set contain 10 and 500 images. Although the validation set has much fewer images than the test set, the open-set classification performances align well on the two sets.

parameter $\lambda_G \in [0.05, 0.10, 0.15, 0.20, \ldots, 0.80, 0.85, 0.90]$. We mainly focus on tuning $\lambda_G$ to study how the synthesized data help training.

In Table 1, we show the performance on the test set of $\text{OpenGAN}^{pix}$ and $\text{OpenGAN}^{fea}$ with varied open training images. For each selected model, we mark the corresponding $\lambda_G$ that yields the best performance (on validation set). Roughly speaking, it is preferable to set a lower weight $\lambda_G$ when we have more real open-set training data. However, we do not see a clear correlation between the weight $\lambda_G$ 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 $\lambda_G$ perform on validation set and test set, and if the model selected on the validation set can reliably perform well on the testing set. Figure 5 plots the performance as a function of $\lambda_G$ on validation set and test set. Hereby we choose the $\text{OpenGAN}^{fea}$ model trained with 1000 open training images. We can see the performance on the validation set reliably reflects the performance on the test set. This confirms that model selection on the validation set is reliable.

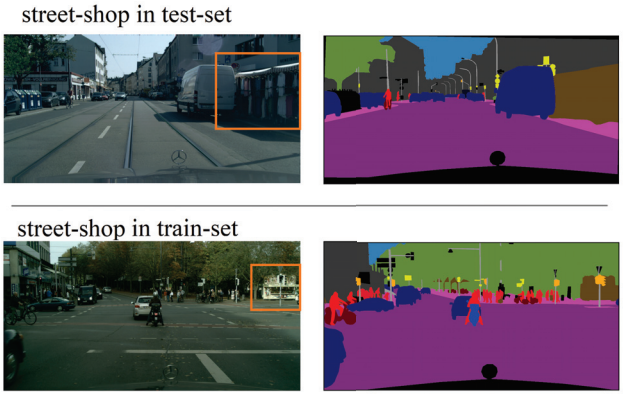


Figure 6: **street-shop as open-set.** Figure 4 in the main paper shows open-set pixel recognition results on a street-shop on a testing image (top-row). We verify if such a street-shop appears in the training set. We manually search for a similar street-shop in the training set, and find the one (bottom-row) 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 testing example shown in the top row. In this sense, the testing image in the top row does contain a real open-set example (i.e., the street-shop) in terms of not only size, but also novel content.

## 5. Statistical Models for Open-Set

Our previous work introduces a lightweight statistical pipeline that repurposes off-the-shef (OTS) deep features for open-set recognition [20]. For the completeness of this paper, we briefly introduce this pipeline: (1) extracting OTS features (with appropriate processing detailed 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 one can choose, e.g., nearest class centroids, Nearest Neighbors, and (class-conditional) Gaussian Mixture Models (GMMs). During testing, we extract the OTS features of the given example and resort to the learned statistical models to compute an open-set likelihood, e.g., based on the (inverse) closed-set probability from GMM. By thresh-

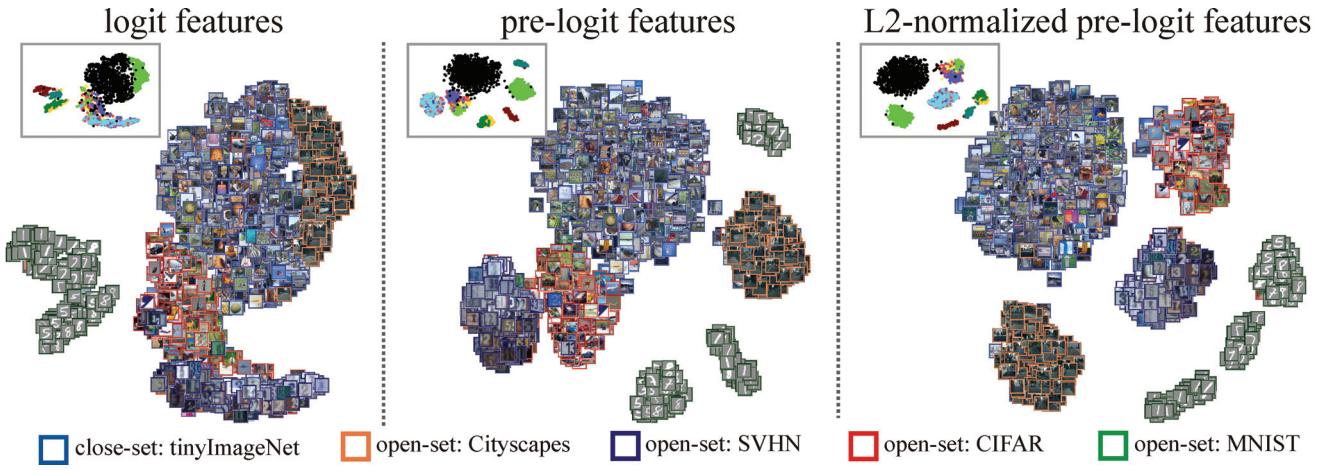


Figure 7: t-SNE plots [23] of open vs closed-set testing data, as encoded by different features from a ResNet18 network trained from scratch on the 200-way TinyImageNet dataset. To better view the clustering results, we zoom-out with scatter plots that color the open-set examples using their class labels provided by the respective datasets (black dots are the closed-set examples of TinyImageNet). **Left**: Logit features mix open and closed data, suggesting that methods based on them (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**: Normalizing the pre-logits features separates them better. These plots intuitively demonstrate the benefit of L2-normalization and using OTS features ranther than the highly-invariant logits.

olding the open-set likelihood, we decide whether it is an open-set example or one of the $K$ closed-set classes, with the latter we report the predicted class label.

**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 [17] and logits [3, 14, 26] 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. 7). One immediate challenge to extract features from an intermediate layer is their high dimensionality, e.g. of size 512x7x7 from ResNet18 [15]. To reduce feature dimension, we simply (max or average) pool the feature activations spatially into a 512-dim feature vectors [30]. 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 [11, 13], we find it crucial to L2-normalize extracted features (Fig. 7). We refer the reader to [20] 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 classes. Such statistical methods include simple parametric models such as class centroids [24] and class-conditional Gaussian models [21, 14], non-parametric models such as NN [5, 19], and mixture models such as (class-conditional) GMMs and k-means [6]. 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. 7). 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 make use of validation set to determine these hyperparameters, as opposed to prior works that conduct model selection either unrealistically on the test-set [26] or on large-scale val-set which could be arguably used for training [21]. We refer the reader to [20] for detailed analysis.

**Lightweight Pipeline**. We re-iterate 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 pool-

ing 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 (such as defined in Cityscapes), it merely 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.

## 6. Further Quantitative Results

While in the main paper we compare OpenGAN to many methods and cannot include more due to space issues, we list a few more in this appendix including Entropy [27], GMM [20], CGDL [28], OpenHybrid [31], and RPL++ [7]. Except for Entropy which is a classic method, the rest were published recently. Table 2 lists the comparisons under *Setup-I*. Please refer to Section 4.2 of the main paper for the detailed setup. Numbers are comparable to Table 1 in the main paper. In summary, our OpenGAN outperforms all these prior methods under this setup, achieving the state-of-the-art.

## 7. Visualization of Generated Images

In this section, we visualize some synthesized examples for intuitive demonstration.

**Generating Small Images.** Recall that OpenGAN-$0^{pix}$ trains a normal GAN and uses its discriminator as the open-set likelihood function. As demonstrated in the main paper, OpenGAN-$0^{pix}$ performs surprisingly well under *Setup-I* (i.e., using CIFAR10, MNIST and SVHN datasets) and *Setup-II* (using TinyImageNet as the closed-set and other datasets as the open-set). OpenGAN-$0^{pix}$ also enables us to generate visual results for intuitive inspection. In Figure 8, we display real and synthesized "fake" images under *Setup-I* on each of the three datasets. In Figure 9, we display real and fake images under *Setup-II* by using tinyImageNet as the closed-set and other datasets as the open-set. We can see the generated images look realistic in terms of color and tone. But they are not strictly open-set images as they contain synthesized *known* contains (e.g., the digits in the synthesized images are closed-set digits). This intuitively demonstrates that a perfectly trained discriminator will not be capable of discriminating open and closed-sets due to the nature of the min-max game in training GANs. However, from the low confidence scores of classifying the generated fake data as closed-set shown in Figure 8, we can see the discriminator almost naively recognizes these synthesized examples as "fake" data. This shows the synthesized data are insufficient to be used for model selection. Moreover, from the classification confidence scores on the closed-testing and open-testing images in each datasets, we can see the discriminator is not calibrated. In other words, we cannot naively set threshold as 0.5 for open-vs-closed classification. This is largely hidden by AUROC metric which is calibration-free. This implies a potential limitation and suggests future work to calibrate the open-set discriminators.

**Generating Cityscapes Patches.** In the main paper (Fig. 6), we have shown some generated patches. In the appendix, we provide more in Figure 10. 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) significantly unperforms OpenGAN-$0^{fea}$ (0.709 AUROC) for open-set segmentation. The "synthesized" patches by OpenGAN-$0^{fea}$ capture many *open-set* objects, such as `bridges`, `vehicle logo` and `back of traffic sign`, all of which are outside the 19 classes defined in Cityscapes. This intuitively explains why OpenGAN-$0^{fea}$ (0.709 AUROC) works much better than OpenGAN-$0^{pix}$ (0.549 AUROC).

## 8. Visual Results of Open-Set Segmentation

On the task of open-set semantic segmentation, first, we show in Figure 6 that our testing set contains real open-set examples never-before-seen in training; please refer to the caption for details. Then, we show more visual results in figures from 11 through 17. From these figures, we can see OpenGAN$^{fea}$ captures most open-set pixels, outperforming the other methods notably.

## 9. Failure Cases and Limitations

As we use an discriminator as the open-set likelihood function, straightforwardly, failure cases happen when the classification is not correct, as shown by the marked confidence scores in Figure 8, as well as the thresholded per-pixel predictions in figures from 11 through 17.

Hereby we point out other failure cases and limitations. First, as we have explained in the main paper, the GAN-discriminator will eventually become incapable of discriminating closed-set and fake/open-set images due to the nature of GANs that strikes an equilibrium between the discriminator and generator. Although we empirically show superior performance by model selection on a validation set,

Table 2: **Open-set discrimination (Setup-I)** measured by area under ROC curve (AUROC)↑. Numbers are comparable to Table 1 in the main paper. Recall that OpenGAN-0 does not train on outlier data (i.e., $\lambda_0$=0 in Eq. 2) and only selects discriminator checkpoints on the validation set. OpenGAN-$0^{fea}$ clearly performs the best, achieving the state-of-the-art.

| Dataset | MSP [17] | Entropy [27] | OpenMax [3] | $MSP_c$ [22] | GOpenMax [10] | OSRCI [25] | MCdrop [9] | GDM [21] | GMM [20] | C2AE [26] | CGDL [28] | RPL-WRN [7] | OpenHybrid [31] | OpenGAN-$0^{fea}$ (*ours*) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *MNIST* | .977 | .988 | .981 | .985 | .984 | .988 | .984 | .989 | .993 | .989 | .994 | .996 | .995 | **.999** |
| *SVHN* | .886 | .895 | .894 | .891 | .896 | .910 | .884 | .866 | .914 | .922 | .935 | .968 | .947 | **.988** |
| *CIFAR* | .757 | .788 | .811 | .808 | .675 | .699 | .732 | .752 | .817 | .895 | .903 | .901 | .950 | **.973** |

there surely exists risks that the validation set is biased in an unknown way which could catastrophically hurt the final open-set recognition performance. This is also true even with outlier training examples. 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 in trainign 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 yet, and we leave it as future work.

## References


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

[2] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*, 2017. 3

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

[4] 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. 2

[5] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *CVPR*. IEEE, 2008. 6

[6] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. Deep quantization network for efficient image retrieval. In *AAAI*, 2016. 6

[7] 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. 7, 8

[8] 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. 2

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

[10] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. In *British Machine Vision Conference (BMVC)*, 2017. 8

[11] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 6

[12] 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. 3

[13] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 2017. 6

[14] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2019. 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[16] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A benchmark for anomaly segmentation. *arXiv preprint arXiv:1911.11132*, 2019. 2

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

[18] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019. 3

[19] Pedro R Mendes Júnior, Roberto M De Souza, Rafael de O Werneck, Bernardo V Stein, Daniel V Pazinato, Waldir R de Almeida, Otávio AB Penatti, Ricardo da S Torres, and Anderson Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017. 6

[20] Shu Kong and Deva Ramanan. An empirical exploration of open-set recognition via lightweight statistical pipelines, 2021. 5, 6, 7, 8

[21] 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. 6, 8

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

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 6

[24] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *ECCV*, 2012. 6


Figure 8: **Demonstration of visuals along with the classification confidence scores as probabilities of being recognized as closed-set data.** On each of the three datasets, we show some random images that are closed-training image (for training GANs), synthesized "fake" images, closed-testing images (from known classes) and open-testing images (from unknown classes). We also mark the probability for each image of being classified as closed-set by the discriminator. We can see the synthesized images look realistic in terms of color, tone and shape. But the discriminator can easily recognize these fake images (as indicated by the low probability). Moreover, although the discriminator achieves good open-vs-closed classification performance measured by AUROC (which is calibration-free), the confidence scores (probability) are not calibrated well. This implies that the discriminator may need to be calibrated for real-world application.

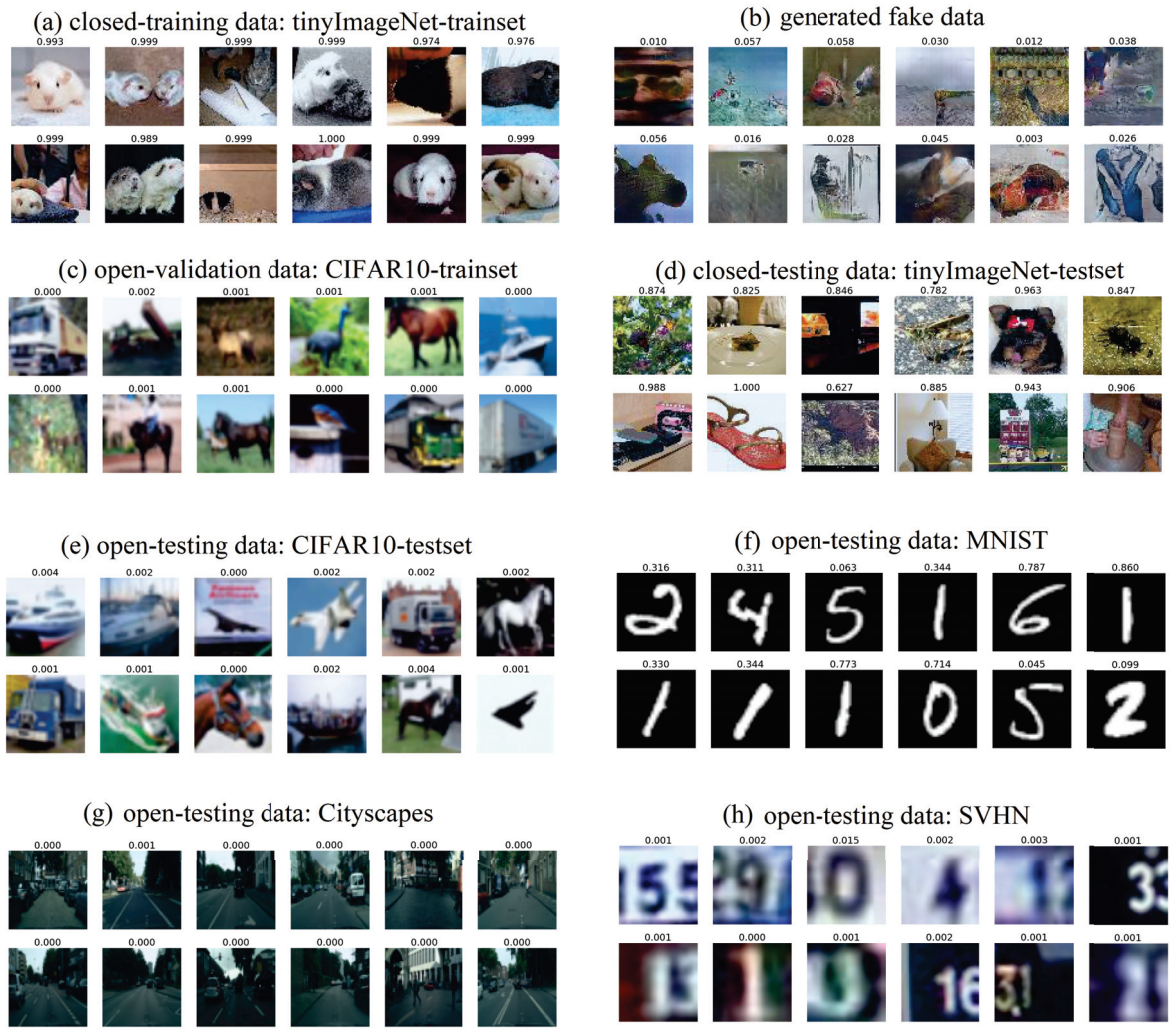


Figure 9: **Demonstration of visuals along with the OpenGAN-0$^{pix}$ classification confidence scores as probabilities of being recognized as closed-set data.** These visual results are generated under *Setup-II*, where the TinyImageNet is the closed-set for 200-way classification, and other datasets are treated as the open-set. The discriminator of the OpenGAN-0$^{pix}$ is seleced over the CIFAR train-set. **(a)** The discriminator recognizes the closed-set training examples with a high confidence score. **(b)** OpenGAN-0$^{pix}$ synthesizes fake images that look realistic in terms of color, tone and shape, but not content. The discriminator can easily recognize these fake images (as indicated by the low probability). The discriminator generalizes well in terms of recognizing closed-set examples from the validation and test sets as shown in **(c)** and **(d)**, and open-set examples from other datasets as shown in **(e)**, **(f)**, and **(h)**.


[25] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *ECCV*, 2018. 8

[26] Poojan Oza and Vishal M. Patel. C2AE: class conditioned auto-encoder for open-set recognition. In *CVPR*, 2019. 6, 8

[27] Jacob Steinhardt and Percy S Liang. Unsupervised risk estimation using only conditional independence structure. In *NeurIPS*, 2016. 7, 8

[28] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *CVPR*, 2020. 7, 8

[29] 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, 3

[30] Songfan Yang and Deva Ramanan. Multi-scale recognition with dag-cnns. In *CVPR*, pages 1215–1223, 2015. 6

[31] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *ECCV*, 2020. 7, 8

[32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2


Figure 10: **Visuals of real Cityscapes image patches (left), synthesized patches by OpenGAN-0$^{pix}$ (mid) and GOpenGAN-0$^{fea}$ (right).** As OpenGAN-0$^{fea}$ generates feature vectors instead of RGB patches, we "synthesize" the patches in an analytical way – for a generated feature, we find the nearest-neighbor per-pixel feature (w.r.t L1 distance) from the training images, and then find the RGB patch centered at the associated pixel with the per-pixel feature. The real patch is our "synthesized" patch for that generated feature. The synthesized patches by OpenGAN-0$^{pix}$ do look realistic in terms of color and tone, but OpenGAN-0$^{pix}$ (0.549 AUROC) does not work as well as OpenGAN-0$^{fea}$ (0.709 AUROC). The "synthesized" patches by OpenGAN-0$^{fea}$ do capture some *unknown open-set* objects, such as `bridge`, `back of traffic sign` and `unknown static objects`, none of which belong to any of the 19 classes defined in Cityscapes for semantic segmentation (cf. Figure 1).

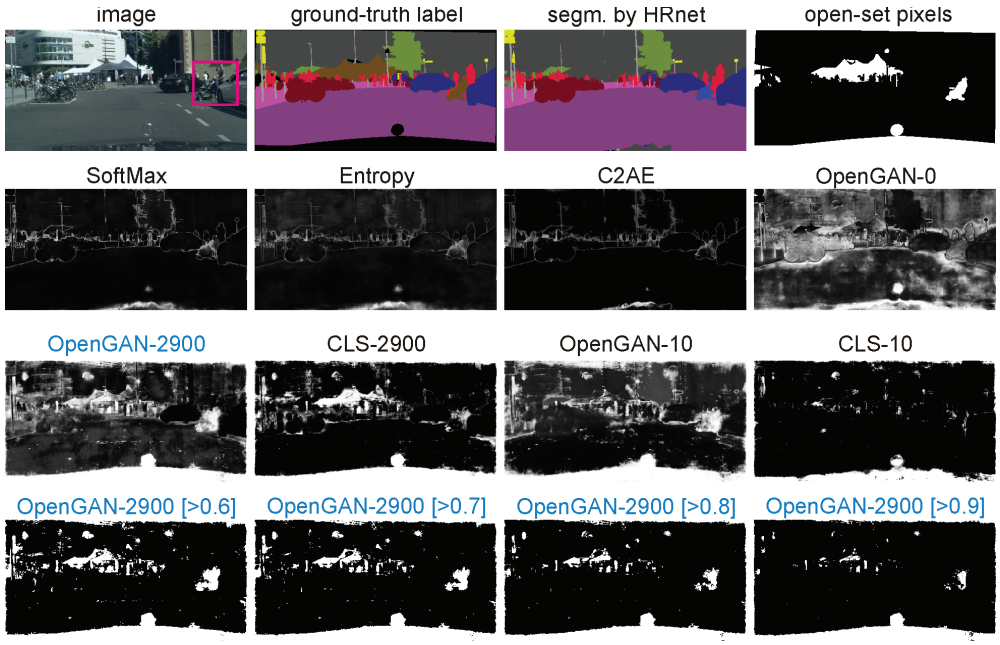


Figure 11: **Qualitative results of a testing image from Cityscapes**. [1$^{st}$ row] the input image, its per-pixel semantic labels, the semantic segmentation result by HRnet and open-set pixels colored by white. [2$^{nd}$ row] visual results as per-pixel scores of being classified as open-set pixel by SoftMax, Entropy, C2AE and our OpenGAN-0$^{fea}$. [3$^{rd}$ row] visual results by our OpenGAN$^{fea}$ and CLS, trained with 2900 or 10 open training images, respectively. [4$^{th}$ row] visual results by thresholding OpenGAN-2900 with 0.6, 0.7. 0.8 and 0.9 respectively. OpenGAN clearly captures most open-set pixels (cf. the white pixels in top-right open-set map).

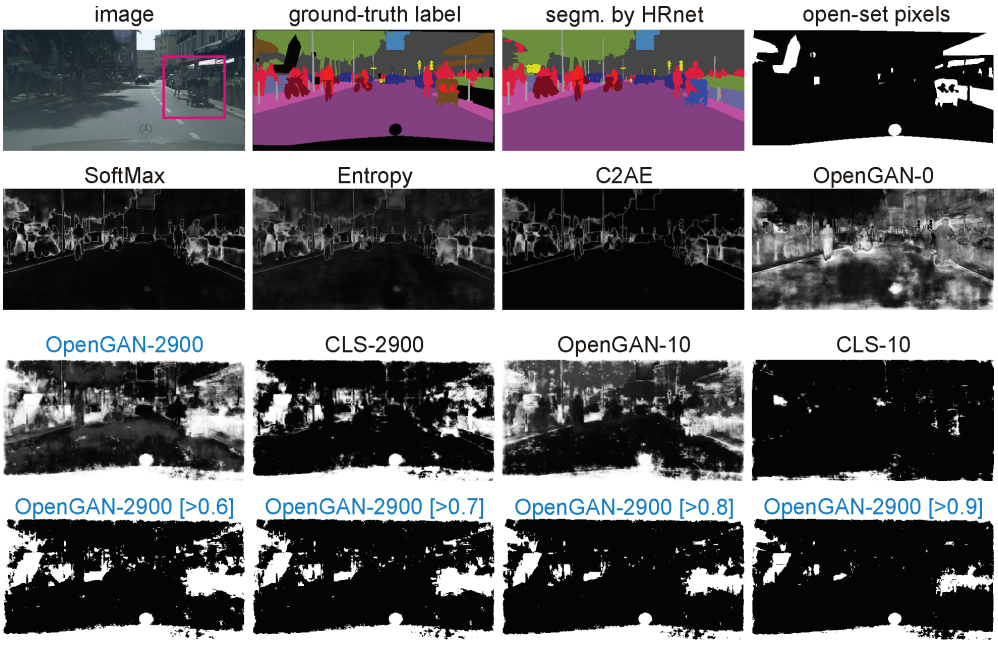


Figure 12: **Qualitative results of a testing image from Cityscapes**. [$1^{st}$ row] the input image, its per-pixel semantic labels, the semantic segmentation result by HRnet and open-set pixels colored by white. [$2^{nd}$ row] visual results as per-pixel scores of being classified as open-set pixel by SoftMax, Entropy, C2AE and our OpenGAN-$0^{fea}$. [$3^{rd}$ row] visual results by our OpenGAN$^{fea}$ and CLS, trained with 2900 or 10 open training images, respectively. [$4^{th}$ row] visual results by thresholding OpenGAN-2900 with 0.6, 0.7. 0.8 and 0.9 respectively. OpenGAN clearly captures most open-set pixels (cf. the white pixels in top-right open-set map).

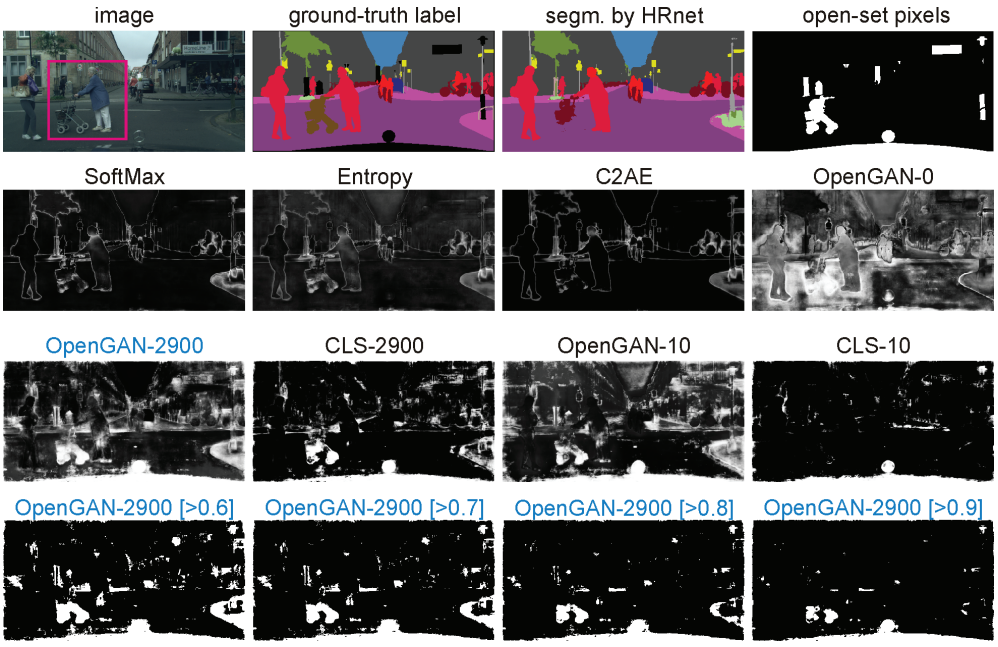


Figure 13: **Qualitative results of a testing image from Cityscapes**. [$1^{st}$ row] the input image, its per-pixel semantic labels, the semantic segmentation result by HRnet and open-set pixels colored by white. [$2^{nd}$ row] visual results as per-pixel scores of being classified as open-set pixel by SoftMax, Entropy, C2AE and our OpenGAN-$0^{fea}$. [$3^{rd}$ row] visual results by our OpenGAN$^{fea}$ and CLS, trained with 2900 or 10 open training images, respectively. [$4^{th}$ row] visual results by thresholding OpenGAN-2900 with 0.6, 0.7. 0.8 and 0.9 respectively. OpenGAN clearly captures most open-set pixels (cf. the white pixels in top-right open-set map).

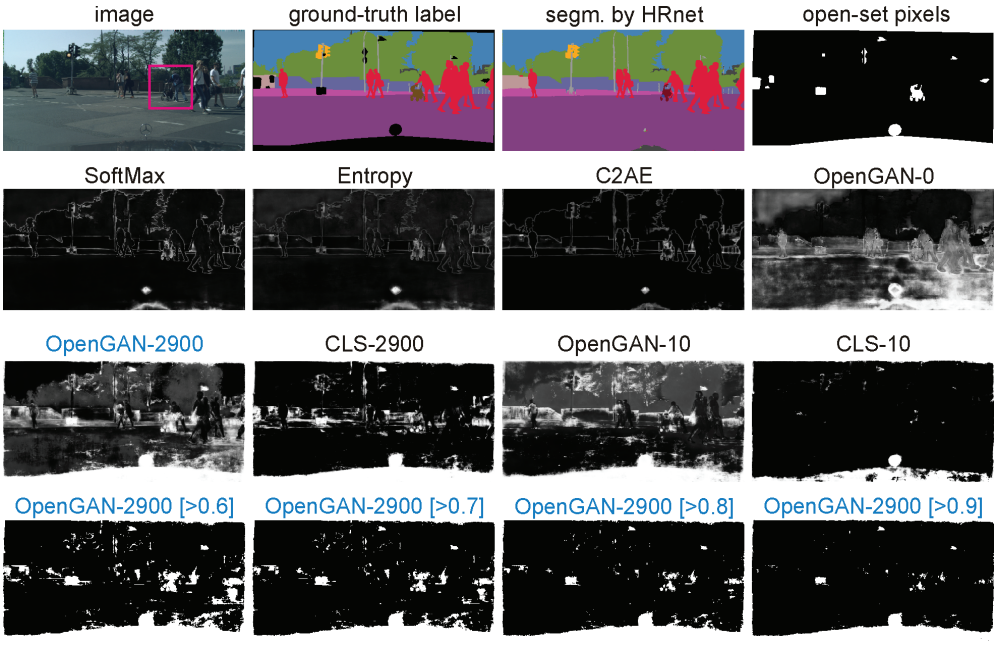


Figure 14: **Qualitative results of a testing image from Cityscapes**. [$1^{st}$ row] the input image, its per-pixel semantic labels, the semantic segmentation result by HRnet and open-set pixels colored by white. [$2^{nd}$ row] visual results as per-pixel scores of being classified as open-set pixel by SoftMax, Entropy, C2AE and our OpenGAN-$0^{fea}$. [$3^{rd}$ row] visual results by our OpenGAN$^{fea}$ and CLS, trained with 2900 or 10 open training images, respectively. [$4^{th}$ row] visual results by thresholding OpenGAN-2900 with 0.6, 0.7. 0.8 and 0.9 respectively. OpenGAN clearly captures most open-set pixels (cf. the white pixels in top-right open-set map).

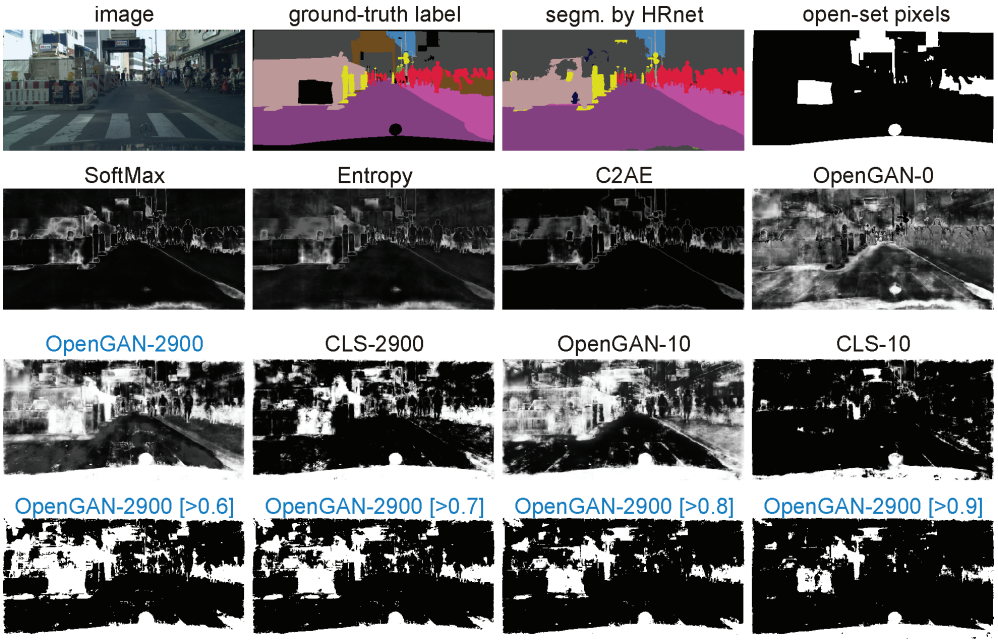


Figure 15: **Qualitative results of a testing image from Cityscapes**. [$1^{st}$ row] the input image, its per-pixel semantic labels, the semantic segmentation result by HRnet and open-set pixels colored by white. [$2^{nd}$ row] visual results as per-pixel scores of being classified as open-set pixel by SoftMax, Entropy, C2AE and our OpenGAN-$0^{fea}$. [$3^{rd}$ row] visual results by our OpenGAN$^{fea}$ and CLS, trained with 2900 or 10 open training images, respectively. [$4^{th}$ row] visual results by thresholding OpenGAN-2900 with 0.6, 0.7. 0.8 and 0.9 respectively. OpenGAN clearly captures most open-set pixels (cf. the white pixels in top-right open-set map).

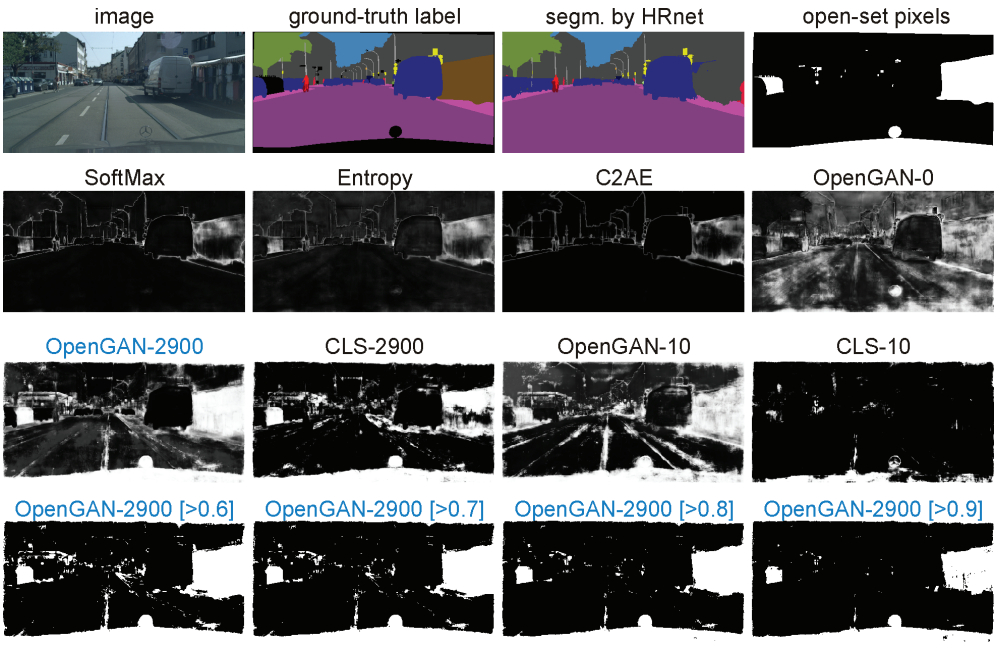


Figure 16: **Qualitative results of a testing image from Cityscapes**. [$1^{st}$ row] the input image, its per-pixel semantic labels, the semantic segmentation result by HRnet and open-set pixels colored by white. [$2^{nd}$ row] visual results as per-pixel scores of being classified as open-set pixel by SoftMax, Entropy, C2AE and our OpenGAN-$0^{fea}$. [$3^{rd}$ row] visual results by our OpenGAN$^{fea}$ and CLS, trained with 2900 or 10 open training images, respectively. [$4^{th}$ row] visual results by thresholding OpenGAN-2900 with 0.6, 0.7. 0.8 and 0.9 respectively. OpenGAN clearly captures most open-set pixels (cf. the white pixels in top-right open-set map).

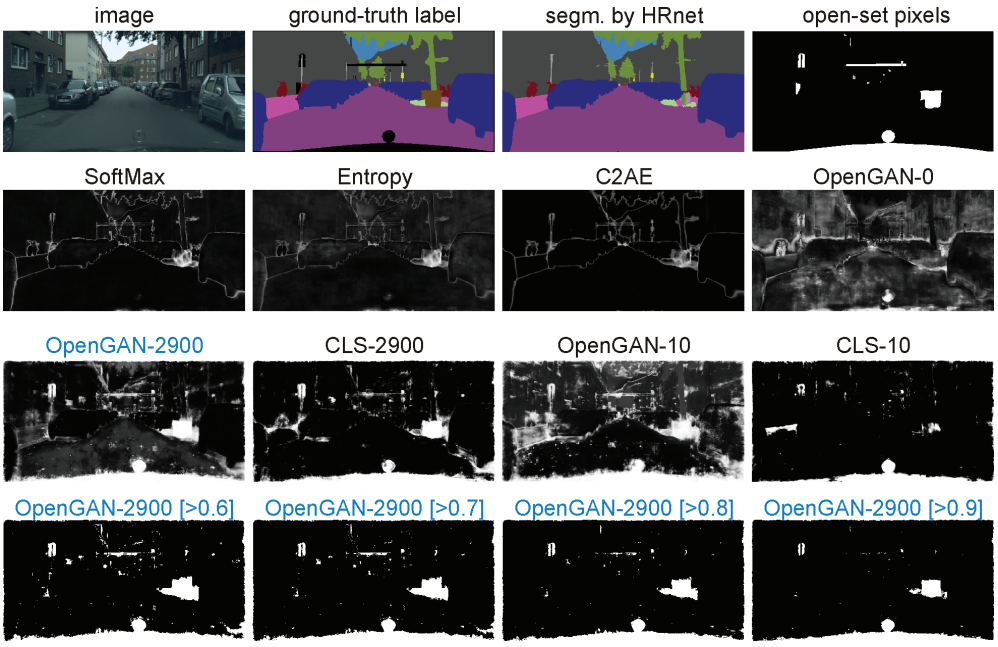


Figure 17: **Qualitative results of a testing image from Cityscapes**. [$1^{st}$ row] the input image, its per-pixel semantic labels, the semantic segmentation result by HRnet and open-set pixels colored by white. [$2^{nd}$ row] visual results as per-pixel scores of being classified as open-set pixel by SoftMax, Entropy, C2AE and our OpenGAN-$0^{fea}$. [$3^{rd}$ row] visual results by our OpenGAN$^{fea}$ and CLS, trained with 2900 or 10 open training images, respectively. [$4^{th}$ row] visual results by thresholding OpenGAN-2900 with 0.6, 0.7. 0.8 and 0.9 respectively. OpenGAN clearly captures most open-set pixels (cf. the white pixels in top-right open-set map).