CODEs: Chamfer Out-of-Distribution Examples against Overconfidence Issue (Supplementary Material)

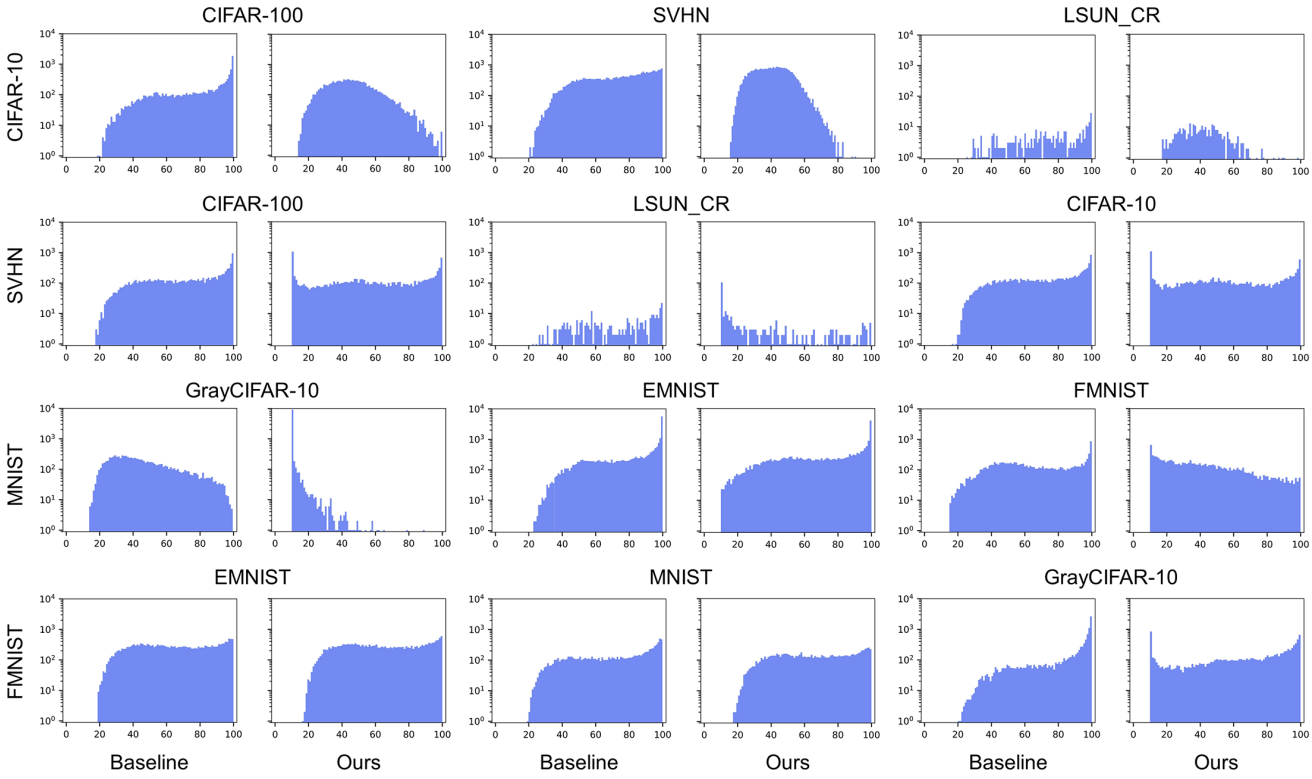

Figure 1: Histograms (logarithmic scale) of maximum confidence values of ResNet-18 trained for CIFAR-10, SVHN, MNIST and FMNIST on various OOD datasets. Axis y denotes the number of samples and axis x denotes the confidence (%).

1. Visualization of Maximum Confidence

We visualize the maximum confidences predicted on the images in various OOD datasets (e.g., MNIST, FMNIST, EMNIST, CIFAR-100, CIFAR-10, SVHN, GrayCIFAR-10, LSUN_CR) by ResNet-18 trained for FMNIST, CIFAR-10, MNIST and SVHN using histograms (logarithmic scale) in Fig. 1. It could be seen that, by adopting our method, the confidence distributions made by Baseline are pulled to the left, with the number of samples in high confidence largely reduced. Particularly, the improvements are more significant on the model trained for CIFAR-10.

2. Ratio of Normal Images to CODEs

We investigate the effects brought by the ratio of normal images to CODEs during the training process by evaluating ResNet-18 trained on SVHN and CIFAR-100. The results in Tab. 1 show that both two ResNet-18 models trained with the ratio of 1:1 obtain the best performance.

3. Chamfer GAN at Different Epochs

We investigate the effects brought by Chamfer GAN trained at different epochs. The results in Tab. 2 show that our method with Chamfer GAN trained at the $800th$ epoch

Ratio		2:1	1:1	1:2	1:4	Ratio		2:1	1:1	1:2	1:4
SVHN	Origin	98.42	**98.45**	98.39	98.42	CIFAR-100	Origin	80.99	82.21	82.83	**83.72**
	CIFAR-10	64.81	**61.09**	64.14	65.72		SVHN	46.58	**44.34**	53.37	52.68
	CIFAR-100	65.33	**54.09**	65.31	66.59		CIFAR-10	53.48	**52.82**	56.49	59.57
	LSUN_CR	36.79	**36.45**	57.14	38.97		LSUN_CR	**49.71**	51.33	51.93	55.68
	Noise	56.57	**35.58**	37.98	46.64		Noise	23.37	19.96	**19.93**	22.29
	Uniform	11.03	**10.34**	17.42	15.13		Uniform	6.12	**1.77**	3.14	3.09

Table 1: The mean maximal confidence (%) of ResNet-18 trained for SVHN and CIFAR-100 with different ratios of normal images to CODEs on various in-distribution and out-of-distribution datasets. Lower value is better except the results on Origin.

Epoch		200	400	800	1200	1600	Epoch		400	1200	1600	1800	2000
SVHN	Origin	98.58	98.51	**98.81**	98.60	98.48	CIFAR-100	Origin	81.87	82.12	81.30	**82.21**	82.19
	CIFAR-10	70.02	69.64	**61.09**	65.81	65.01		SVHN	47.80	45.85	45.74	**44.34**	44.89
	CIFAR-100	70.39	68.44	**54.09**	57.03	57.59		CIFAR-10	55.85	54.57	54.03	**52.82**	53.23
	LSUN_CR	69.87	47.70	**36.45**	46.89	65.87		LSUN_CR	53.84	53.66	52.72	**51.33**	51.39
	Noise	45.39	37.76	**35.58**	44.92	42.86		Noise	21.90	23.11	20.81	**19.96**	22.89
	Uniform	10.88	12.33	**10.34**	11.37	14.61		Uniform	9.34	4.71	2.32	**1.77**	2.67

Table 2: The mean maximal confidence (%) of ResNet-18 trained for SVHN and CIFAR-100 on various in-distribution and out-of-distribution datasets with Chamfer GAN trained at different epochs. Lower value is better except the results on Origin.

Method		Baseline	+Seeds	+CODEs	Method		Baseline	+Seeds	+CODEs
CIFAR-10	ResNet-20	92.54	92.67	**92.98**	CIFAR-100	ResNet-20	67.11	67.47	**67.69**
	ResNet-56	93.53	93.63	**93.89**		ResNet-56	72.23	72.58	**73.09**
	WRN-28-10	95.65	95.73	**96.03**		WRN-28-10	80.14	80.55	**81.08**
	GoogLeNet	94.89	95.09	**95.29**		GoogLeNet	79.91	80.19	**80.92**

Table 3: Top-1 classification accuracy (%) of ResNet-18 on CIFAR-10 and CIFAR-100. Larger value is better.

performs the best on SVHN, since with limited amount of training data, while the performance on CIFAR-100 keeps improving until the $1800th$ epoch.

4. Improving Classification

We demonstrate utilizing seed examples and CODEs to improve classification. Specifically, we train multiple different network architectures with suppressing predictions on seed examples and CODEs using a separate batch norm following [5]. The results in Tab. 3 show that training with suppressing predictions on seed examples could bring a certain amount of improvement already, and the improvement is further enlarged when we adopt CODEs, validating the usefulness of CODEs in improving classification and the importance of Chamfer GAN.

5. Detecting Semantic OOD Examples

We train ResNet-18 classifiers for CIFAR-10 with holding out one class every time, and then score the ability to detect the held out class as OOD samples following [1], such that in-distribution samples are not only significantly outnumber OOD ones, but also have significant semantic shifts. The precision-recall (PR) curves are presented in Fig. 2. It could be seen that, OE [3] and CCUd [4], both of which adopt auxiliary datasets, hurt the performance of semantic OOD detection, while utilizing CODEs reinforces it.

6. Implementation Details

Experimental Platform. We implement all the models in PyTorch and train them on NVIDIA Tesla V100 GPUs.

Chamfer GAN. Denote $c(i, j, k)$ as a convolution layer, where each convolution uses kernel of size $i \times i$, with a

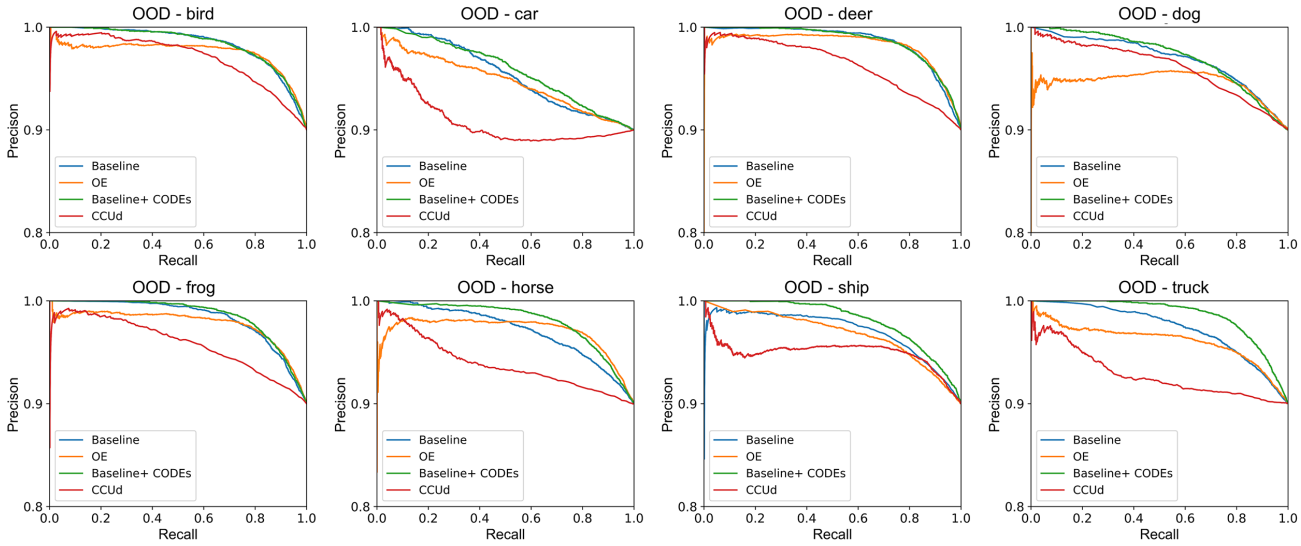

Figure 2: PR curves of four methods for the semantic OOD detection task [1] on CIFAR-10 with holding out one class as OOD.

stride of j, and a padding of k, and let $C(i, j, k)$ be a group of layers with $c(i, j, k)$, BatchNorm (BN) and ReLU. For images with the resolution of 32×32, our encoder in Chamfer GAN consists of the following four groups: $C(4, 2, 1) - C(4, 2, 1) - C(4, 2, 1) - C(4, 2, 1)$, with the channel numbers: 3-64-128-256-512. For images with the resolution of 28×28, we replace the encoder as $C(4, 2, 1) - C(4, 2, 2) - C(4, 2, 1) - C(4, 2, 1)$. The decoder in Chamfer GAN is symmetric to the encoder, except replacing the convolutions with transposed convolutions. Note that, we do not intentionally design the structure of auto-encoder.

OOD Tasks. We use ADAM on MNIST with a learning rate of $1e$-3 and SGD with learning rate 0.1 for the other datasets. We decrease all learning rates by a factor of 10 after 50, 75 and 90 epochs for a total of 100 epochs. The batch size is set to 128, and weight decay is set to $5e$-4. We set the ratio of normal images to seed examples, CODEs and samples in auxiliary datasets to be 1:1. The settings are the same as in [4] for fair comparisons.

Classification Tasks. All the models are trained from scratch with SGD using default parameters as the optimizer, and the weights are initialized following [2].

For the models on CIFAR-10 and CIFAR-100, we set the initial learning rate with 0.1, and divide it by 5 at 60, 120 and 160 epochs for total 200 epochs. For data augmentation, we pad 4 pixels on each side of the image, and randomly sample a 32×32 crop from the padded image or its horizontal flip, and then apply the simple mean/std normalization.

For the models on CINIC-10, we train the models on the train set with a mini-batch of 128 and evaluate them on the test set. The training starts with an initial learning rate of 0.1, and cosine annealed to zero for total 300 epochs, based on the same data augmentation scheme as in CIFAR datasets.

For the models on ImageNet, we train the models with data augmentation including random resized crop, flip and mean/std normalization on the training set with a mini-batch size of 256, and report results on the validation set. The initial learning rate is set to 0.1 and decreased by a factor of 10 every 30 epochs to a total of 100 epochs.

7. More Visualizations

Fig. 3 and Fig. 4 visualize more seed examples and their corresponding CODEs of ImageNet and CIFAR-10.

References

[1] Faruk Ahmed and Aaron Courville. Detecting semantic anomalies. In *AAAI*, volume 34, pages 3154–3162, 2020. 2, 3

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[3] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2018. 2

[4] Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. In *ICLR*, 2020. 2, 3

[5] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *CVPR*, pages 819–828, 2020. 2

Figure 3: Seed examples and the corresponding CODEs of ImageNet.

Figure 4: Seed examples and the corresponding CODEs of CIFAR-10.