Supplementary Material
AdaptiveMix: Robust Feature Representation via Shrinking Feature Space

Haozhe Liu[1†], Wentian Zhang[2†], Bing Li[1✉], Haoqian Wu[3], Nanjun He[2], Yawen Huang[2]
Yuexiang Li[2✉], Bernard Ghanem[1], Yefeng Zheng[2]
[1] AI Initiative, King Abdullah University of Science and Technology (KAUST),
[2]Jarvis Lab, Tencent, [3]YouTu Lab, Tencent
{*haozhe.liu;bing.li;bernard.ghanem*}@*kaust.edu.sa; zhangwentianml@gmail.com;*
{*linuswu;yawenhuang;nanjunhe;vicyxli;yefengzheng*}@*tencent.com*

Abstract

In this supplementary material, we first provide the implementation details of integrating our AdaptiveMix with image generation methods, as well as that of applying it to various visual recognition tasks such as robust image recognition and OOD detection in Sec A.

We then elaborate on additional experimental details, including datasets and additional evaluation criteria for the image generation task in Sec. B. More experimental results on image generation are also presented, showing our AdaptiveMix effectively improves existing state-of-the-art image generation methods. Besides, we provide additional experimental details of our AdaptiveMix on robust image recognition and Out-Of-Distribution (OOD) tasks in Secs. C, D and E.

A. Implementation Details

In this section, we elaborate on the implementation detail of our proposed method. The public platform PyTorch [35] is used to implement the experiments. Our models are trained on a workstation with a CPU of 2.8GHz, RAM of 512GB, and 8 GPUs of NVIDIA Tesla V100 with 32 GB memory capacity.

Our method is a plug-and-play module that can be integrated with different methods for various tasks. Hence, the training strategies depend on the integrated method and the task. In our main paper, we first integrate our proposed AdaptiveMix with image generation methods *i.e*. WGAN and StyleGAN-V2, respectively, and then apply it to recognition tasks, including image classification, robust image recognition, and OOD detection. To provide implementation details for our main paper, we first show the pseudo-code for the proposed method on WGAN and then present our training strategy on StyleGAN-V2. Finally, we summarize the pseudo-code of AdaptiveMix for visual recognition.

AdaptiveMix-based Image Generation. To comprehensively evaluate AdaptiveMix, we respectively integrate it with two state-of-the-art image generation methods, WGAN [1] and StyleGAN-V2 [20], namely "AdaptiveMix-based WGAN" and "AdaptiveMix-based StyleGAN-V2". The pseudo-code of "AdaptiveMix-based WGAN" is summarized in Algorithm 1. Firstly, AdaptiveMix generates hard samples $\hat{x}$ by the convex combination of real samples x and generated samples x_g. Then, AdaptiveMix loss is integrated into the final learning objective. Sec. B provides additional experimental details about our AdaptiveMix-based image generation method.

Different from the training of WGAN, StyleGAN-V2 coupled with a series of advanced components for the unsupervised image generation, including Style Mixing [19] and Path Length Regularization. To avoid the disruption of its original stable training by directly using hard samples, We modify the proposed AdaptiveMix to be used to train StyleGAN-V2. As shown in Algorithm 2, we employ AdaptiveMix to the real and generated samples separately.

AdaptiveMix-based Visual Recognition. The pseudo-code of AdaptiveMix for visual recognition is summarized in Algorithm 3. The final learning objective contains the mixup-based cross-entropy loss function and the proposed AdaptiveMix loss. Then, the trained network with AdaptiveMix is used for image classification and the Out-Of-Distribution (OOD) detection in this paper.

B. Additional Details on Image Generation

This section introduces the datasets and experimental settings for image generation, adversarial attack defense, and OOD detection tasks, respectively.

Algorithm 1 AdaptiveMix-based WGAN

Input:
Generator $G_\theta(\cdot)$; Feature Extractor $\mathcal{F}_\gamma(\cdot)$; Classifier Head $\mathcal{J}_\beta(\cdot)$; The number of critic iterations per generator iteration n_c
Output:
Trained Parameters θ;

while θ has not converged **do**
  for $t = 1$ to n_c **do**
    Sample $x \sim p_r$, latent variable $z \sim p_z$;
    Sample λ from Beta distribution $\mathbb{B}(\alpha, \alpha)$;
    $x_g \leftarrow G_\theta(z)$;
    $\hat{x} \leftarrow g(x, x_g, \lambda)$ by Eq. (1);
    $\mathcal{L}_{wgan} \leftarrow \mathbb{E}_{z \sim p_z}[\mathcal{J}(\mathcal{F}(G(z)))] - \mathbb{E}_{x \sim p_r}[\mathcal{J}(\mathcal{F}(x))]$;
    $\mathcal{L} \leftarrow \mathcal{L}_{wgan} + \mathbb{E}_{x \sim p_r, p_g}[\mathcal{L}_{ada}]$;
    $\gamma_t \leftarrow$ Adam$(\frac{\partial \mathcal{L}}{\partial \gamma_{t-1}})$;
    $\beta_t \leftarrow$ Adam$(\frac{\partial \mathcal{L}}{\partial \beta_{t-1}})$;
  end for
  Sample latent variable $z \sim p_z$;
  $\mathcal{L} \leftarrow \mathbb{E}_{x \sim p_r, p_g}[\mathcal{L}_{ada}] - \mathbb{E}_{z \sim p_z}[\mathcal{J}(\mathcal{F}(G(z)))]$;
  $\theta \leftarrow$ Adam$(\frac{\partial \mathcal{L}}{\partial \theta})$;
end while
Return θ;

Algorithm 2 AdaptiveMix-based StyleGAN-V2

Input:
Generator $G_\theta(\cdot)$; Feature Extractor $\mathcal{F}_\gamma(\cdot)$; Classifier Head $\mathcal{J}_\beta(\cdot)$; The number of critic iterations per generator iteration n_c
Output:
Trained Parameters θ;

while θ has not converged **do**
  for $t = 1$ to n_c **do**
    Sample $x_i, x_j \sim p_r$, latent variable $z_i, z_j \sim p_z$;
    Sample λ from Beta distribution $\mathbb{B}(\alpha, \alpha)$;
    $x_{gi} \leftarrow G_\theta(z_i)$; $x_{gj} \leftarrow G_\theta(z_j)$;
    $\hat{x} \leftarrow g(x_i, x_j, \lambda)$;
    $\hat{x}_g \leftarrow g(x_{gi}, x_{gj}, \lambda)$;
    $\mathcal{L}_g \leftarrow \mathbb{E}_{\hat{x}_g \sim p_g}[\mathcal{J}(\mathcal{F}(\hat{x}_g))] - \mathbb{E}_{\hat{x} \sim p_r}[\mathcal{J}(\mathcal{F}(\hat{x}))]$;
    $\mathcal{L} \leftarrow \mathcal{L}_g + \mathbb{E}_{\hat{x} \sim p_r}[\mathcal{L}_{ada}] + \mathbb{E}_{\hat{x}_g \sim p_g}[\mathcal{L}_{ada}] + R_1$ Reg.;
    $\gamma_t \leftarrow$ Adam$(\frac{\partial \mathcal{L}}{\partial \gamma_{t-1}})$;
    $\beta_t \leftarrow$ Adam$(\frac{\partial \mathcal{L}}{\partial \beta_{t-1}})$;
  end for
  Sample latent variable $z \sim p_z$;
  $\mathcal{L} \leftarrow \mathbb{E}_{\hat{x}_g \sim p_g}[\mathcal{L}_{ada}] + PL$ Reg. $- \mathbb{E}_{z \sim p_z}[\mathcal{J}(\mathcal{F}(\hat{x}_g))]$;
  $\theta \leftarrow$ Adam$(\frac{\partial \mathcal{L}}{\partial \theta})$;
end while
Return θ;

Algorithm 3 AdaptiveMix-based Visual Recognition

Input:
Feature Extractor $\mathcal{F}(\cdot)$; Orthogonal Classifier $\tilde{\mathcal{J}}(\cdot)$;
Output:
Trained $\mathcal{F}(\cdot)$;

Initialize $\tilde{\mathcal{J}}(\cdot)$ through Eq. (9);
while $\mathcal{F}(\cdot)$ has not converged **do**
  Sample $(x_i, y_i), (x_j, y_j) \sim (\mathcal{X}, \mathcal{Y})$;
  Sample λ from Beta distribution $\mathbb{B}(\alpha, \alpha)$;
  $\hat{x}_{ij} \leftarrow g(x_i, x_j, \lambda)$ by Eq. (1);
  $\hat{y}_{ij} \leftarrow g(y_i, y_j, \lambda)$ by Eq. (1);
  $v_i, v_j, \hat{v}_{ij} \leftarrow \mathcal{F}(x_i), \mathcal{F}(x_j), \mathcal{F}(\hat{x}_{ij})$;
  $\mathcal{L}_c \leftarrow \hat{y}_{ij} log(\tilde{\mathcal{J}}(\mathcal{F}(\hat{x}_{ij}))) + \hat{y}_{ij} log(\tilde{\mathcal{J}}(g(v_i, v_j, \lambda)))$
  $\mathcal{L}_{ada} \leftarrow Eq.(2)$
  $\mathcal{L}_t \leftarrow \mathcal{L}_c + \mathcal{L}_{ada}$;
  Update $\mathcal{F}(\cdot)$ by minimizing $\mathcal{L}_t$;
end while
Return $\mathcal{F}(\cdot)$;

B.1. Datasets and Experimental Settings

Synthetic Dataset consists of data from two different distributions, including mixed Gaussian distribution [34] and mixed circle lines [3]. 50k points are sampled from the distribution and each point is represented as a vector containing abscissa and ordinate values. $G(\cdot)$ consists of 4 fully-connected hidden layers and $D(\cdot)$ is composed of three fully-connected layers. ReLU activation and batch normalization are used in $G(\cdot)$. The input code z is a 32-dimensional vector sampled from a standard normal distribution. Models are trained by Adam [22] for 500 epochs.

CIFAR10 [23]. For this dataset, DCGAN [36] is selected as the architecture to test the performance of different learning objectives. The model is trained by Adam with β_1=0.0 and β_2=0.999. The learning rate is 0.0001, with a decay rate of 0.9 for every 50 epochs. The batch size for training is 64. A 64-dimensional Gaussian distribution is adopted as the input for $G(\cdot)$, while the output of $f(D(\cdot))$ is set as a 16-dimensional embedding code.

CelebA [27]. The images are cropped, aligned, and resized to 256×256. The learning rate is 0.0001 with a decay rate of 0.9 per 2 epochs. A 128-dimensional Gaussian distribution is adopted as the input for $G(\cdot)$, and the output of $f(D(\cdot))$ is set as a 32-dimensional embedding code. $D(\cdot)$ and $G(\cdot)$ are updated step by step. The remaining settings, including architecture, optimizer, and evaluation metric, are identical to the setting for CIFAR10.

AFHQ-CAT [5] includes 5,153 closeups for cat faces. We resized all images to the resolution of 256×256 using a high-quality Lanczos filter [24]. In this case, StyleGAN-

V2 [20] is set as the baseline. We kept its details identical with ADA [18], such as network architectures [20], weight demodulation [20], style mixing regularization [19], path length regularization, lazy regularization [20], equalized learning rate for all trainable parameters [17], non-saturating logistic loss [8] with R_1 regularization [32] and Adam optimizer [22].

FFHQ [19] consists of 70,000 human face images. We used a downscaled 256 $\times$ 256 version of FFHQ for training. We also applied a subset of FFHQ, *i.e.* FFHQ-5k [19], which only contains 5,000 images for further discussion. StyleGAN-V2 [20] is set as the baseline in this case and all settings are identical to the setting for AFHQ-CAT.

Baselines. Since the proposed AdaptiveMix does not focus on new network architecture but introduces objective functions, we mainly compare the proposed method to other popular objectives for GANs, including standard GAN(Std-GAN) [8], WGAN [1], WGAN-GP [10], HingeGAN [51], LSGAN [31] and Realness GAN [44]. To further evaluate our method on image generation task, we integrate AdaptiveMix on StyleGAN-V2 [20] and compare it to other regularization methods for GAN training, including Instance Noise [40], One-sided LS [38], LC-Reg [41], ADA [18] and APA [16].

Additional Evaluation Criterion. To quantify the generation performance of the different methods, Fréchet Inception Distance (FID) [15] and Inception Score (IS, higher is better) [38] are adopted as the metrics. In all experiments, 50,000 images are randomly sampled to calculate FID and IS. To evaluate the connection between the proposed AdaptiveMix and Lipschitz continuity, we design a metric as follows:

$$Lip_c = \frac{1}{n}\sum_{i,j\in n}\frac{\mathbb{D}_v(\mathcal{F}(x_i),\mathcal{F}(x_j))}{\mathbb{D}_x(x_i,x_j)} \tag{1}$$

where x_i, x_j are the given pairs of samples. $\mathcal{F}(\cdot)$ is the discriminator of GAN. $\mathbb{D}_v(\cdot)$ calculates the distance between two embedding features and averages them along the feature dimension. $\mathbb{D}_x(\cdot)$ calculates the distance between two images and also averages them to a value. The smaller Lip_c is the better performance of $\mathcal{F}(\cdot)$ to guarantee that the Lipschitz continuity can be achieved.

B.2. Additional Experimental Results

Fig. A shows the FID convergence curves of WGAN, WGAN-GP and AdaptiveMix (Ours), demonstrating that our AdaptiveMix method improves the training convergence substantially compared with WGAN.

Fig. B shows examples of the generated CIFAR-10 and CelebA images when using AdaptiveMix in DCGAN. We

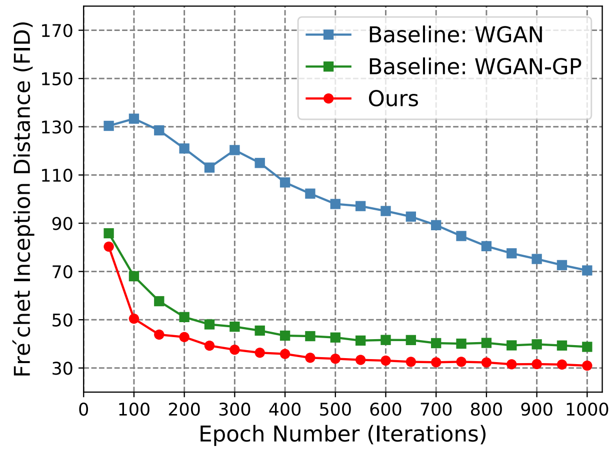

Figure A. Training curves of WGAN, WGAN-GP and AdaptiveMix (Ours) on CIFAR10.

can see that AdaptiveMix can yield comparable results. As shown in Fig. C, we show more generated images for the FFHQ dataset. The images were generated with truncation ϕ=0.75 and selected by setting random seeds. We can see that StyleGAN-V2 with AdaptiveMix can produce high-quality and photorealistic human faces.

C. Additional Details on Clean Image Recognition

Datasets and Experimental Settings. The effectiveness of AdaptiveMix on image recognition is evaluated on CIFAR-10, CIFAR100 [23], Tiny ImageNet [6] and ImageNet [37]. The WideResNet [49] with a depth of 28 and width of 10 (WRN-28-10) is adopted as the backbone for CIFAR-10/100. For the Tiny-ImageNet [6], the backbone is set as PreActResNet18 [13]. While for ImageNet [37], we used the ResNet [12] with a depth of 50 (ResNet-50) as the backbone. In particular, the models are trained using SGD with a weight decay of 0.0005 and a momentum of 0.9. For CIFAR-10/100 and Tiny ImageNet, models are observed to converge after 200 epochs of training. The list of learning rates is set to [0.1, 0.02, 0.004, 0.0008], in which the learning rate decreases to the next after every 60 training epochs. The noise term σ is set to 0.05 for CIFAR-10 and 0.005 for CIFAR-100, respectively. For ImageNet, ResNet-50 is trained for 90 epochs using downscaled 128 $\times$ 128 resolution images as input. The learning rate starts from 0.1 and decays at 0.1 per 30 epochs.

D. Additional Details on Robust Image Recognition

Datasets and Experimental Settings. The robustness of the proposed method against adversarial attacks is evalu-

(a) CIFAR10

(b) CelebA

Figure B. Generated images for (a) CIFAR-10 and (b) CelebA on the real dataset using DCGAN architecture with AdaptiveMix

Figure C. The experimental results carried on the FFHQ. The images correspond to random output produced by the generator of StyleGAN-V2 with the proposed AdaptiveMix when truncation using ϕ=0.75

ated on CIFAR-10, CIFAR100 [23], and Tiny ImageNet [6]. The training strategy and backbones for robust image recognition are identical to Sec C. For data augmentation, we employ horizontal flipping and cropping from the image padded by four pixels on each side in this experiment. To guarantee the fairness of performance comparison, all the experiments are conducted under the same training protocol.

Two interpolation-based methods, *i.e.,* Mixup [50] and Manifold-Mixup [42], are involved for comparison in this study. Although there are recent papers proposing new ways to mix samples in the input space [11, 21, 47], they do not achieve significant improvements over Mixup or Manifold-Mixup, especially against adversarial attacks [21]. Therefore, Mixup and Manifold-Mixup remain the most relevant competing methods among the zoo of Mixup. Note that our mixing strategy is based on Manifold-Mixup, which performs as a solid baseline to validate the effectiveness of AdaptiveMix. For a more comprehensive analysis of the proposed method, an effective adversarial training, free-AT [39], is also included for reference as the upper bound. The evaluation metric is the classification accuracy on the whole test set.

Attack Methods. To evaluate the robustness against adversarial attacks, three popular adversarial attack methods, including FGSM [9], PGD [2, 30] and CW [4], are involved in this study. The perturbation budget is set to 8/255 and 4/255 under l_∞ norm distance for single- and multi-step attacks. PGD-K denotes a K-step attack with a step size of 2/255. For CW, two cases are considered, in which the steps are set to 100 steps, and c is set to 0.01 and 0.05, respectively.

E. Additional Details on OOD Detection

Datasets and Experimental Settings In the OOD detection scenario, the training set of CIFAR-10 [23] is adopted as the in-distribution data, and the test set of CIFAR-10 refers to the positive samples for OOD detection. Similar to the prior works [25, 26, 28, 48], the OOD datasets include Tiny-ImageNet [6] and LSUN [46]. Tiny-ImageNet (a subset of ImageNet [6]) consists of 10,000 test images with a size of 36 $\times$ 36 pixels, which can be categorized into 200 classes. LSUN [46] consists of 10,000 test samples from 10 different scene groups. Since the image size of Tiny-ImageNet and LSUN are not identical to that of CIFAR-10, two downsampling strategies (crop (C) and resize (R)) are adopted for image size unification, following the protocol of [26, 43, 48]. Therefore, we have four OOD test datasets, *i.e.,* TIN-C, TIN-R, LSUN-C, and LSUN-R. The training protocol and backbone for OOD detection is identical to Sec. D.

For the competing methods, Softmax Pred. [14], Counterfactual [33], CROSR [45], OLTR [28] and Union of 1D Subspaces [48] are included. We also exploit the solutions using Monte Carlo sampling or OOD samples [25, 26] as the references for competing methods. To some extent, these methods can be seen as the upper bound for OOD detection regardless of time consumption and over-fitting. For example, Monte Carlo sampling [7, 29] could generally yield improvements to most current OOD methods with huge extra computational costs. The evaluation metric for OOD detection is the F1 score, *i.e.,* the maximum score over all possible threshold ϕ^*.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017. 1, 3

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018. 4

[3] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and J"orn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019. 2

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 4

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 3, 4, 5

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016. 5

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27:2672–2680, 2014. 3

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. 4

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. 3

[11] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3714–3722, 2019. 4

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 3

[14] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations*, 2017. 5

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[16] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Deceive d: Adaptive pseudo augmentation for gan training with limited data. *Advances in Neural Information Processing Systems*, 34:21655–21667, 2021. 3

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations*, 2018. 3

[18] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 3

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 3

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 3

[21] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020. 4

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2, 3

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report TR-2009, University of Toronto, Toronto*, 2009. 2, 3, 4, 5

[24] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950. 2

[25] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018. 5

[26] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *International Conference on Learning Representations*, 2018. 5

[27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 2

[28] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 5

[29] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for Bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32:13153–13164, 2019. 5

[30] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. 4

[31] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2794–2802, 2017. 3

[32] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 3

[33] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European conference on computer vision*, pages 613–628, 2018. 5

[34] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017. 2

[35] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 1

[36] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 3

[38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29, 2016. 3

[39] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 2019. 4

[40] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *International Conference on Learning Representations*, 2017. 3

[41] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7921–7931, 2021. 3

[42] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019. 4

[43] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European conference on computer vision*, pages 550–564, 2018. 5

[44] Yuanbo Xiangli, Yubin Deng, Bo Dai, Chen Change Loy, and Dahua Lin. Real or not real, that is the question. *International Conference on Learning Representations*, 2020. 3

[45] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4016–4025, 2019. 5

[46] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[47] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 4

[48] Alireza Zaeemzadeh, Niccolò Bisagno, Zeno Sambugaro, Nicola Conci, Nazanin Rahnavard, and Mubarak Shah. Out-of-distribution detection using union of 1-dimensional subspaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9452–9461, 2021. 5

[49] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*. British Machine Vision Association, 2016. 3

[50] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4

[51] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. In *International Conference on Learning Representations*, 2017. 3