DOING MORE WITH LESS: IMPROVING ROBUSTNESS USING GENERATED DATA

Sven Gowal*, Sylvestre-Alvise Rebuffi*, Olivia Wiles, Florian Stimberg, Dan Calian and **Timothy Mann**
DeepMind, London
{sgowal,sylvestre}@google.com

ABSTRACT

Recent work argue that robust training requires substantially larger datasets than those required for standard classification. On CIFAR-10 and CIFAR-100, this translates into a sizable robust-accuracy gap between models trained solely on data from the original training set and those trained with additional data extracted from the "80 Million Tiny Images" dataset (80M-TI). In this paper, we explore how state-of-the-art generative models can be leveraged to artificially increase the size of the original training set and improve adversarial robustness to ℓ_p-norm bounded perturbations. We demonstrate that it is possible to significantly reduce the robust-accuracy gap to models trained with additional real data. Surprisingly, we also show that even the addition of non-realistic random data (generated by Gaussian sampling) can improve robustness. We evaluate our approach on CIFAR-10 and CIFAR-100 against ℓ_∞ and ℓ_2 norm-bounded perturbations of size $\epsilon = 8/255$ and $\epsilon = 128/255$, respectively. We show large absolute improvements in robust accuracy compared to previous state-of-the-art methods. Against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$, our model achieves 63.58% and 33.49% robust accuracy on CIFAR-10 and CIFAR-100, respectively (improving upon the state-of-the-art by +6.44% and +3.29%). Against ℓ_2 norm-bounded perturbations of size $\epsilon = 128/255$, our model achieves 78.31% on CIFAR-10 (+3.81%). These results beat most prior works that use external data.

1 INTRODUCTION

Neural networks are being deployed in a wide variety of applications ranging from ranking content on the web (Covington et al., 2016) to autonomous driving (Bojarski et al., 2016) via medical diagnostics (De Fauw et al., 2018). It has become increasingly important to ensure that deployed models are robust and generalize to various input perturbations. Unfortunately, the addition of imperceptible adversarial perturbations can cause neural networks to make incorrect predictions (Carlini & Wagner, 2017a;b; Goodfellow et al., 2015; Kurakin et al., 2016; Szegedy et al., 2014). There has been a lot of work on understanding and generating adversarial perturbations (Szegedy et al., 2014; Carlini & Wagner, 2017b; Athalye & Sutskever, 2018), and on building defenses that are robust to such perturbations (Goodfellow et al., 2015; Madry et al., 2018; Zhang et al., 2019; Rice et al., 2020).

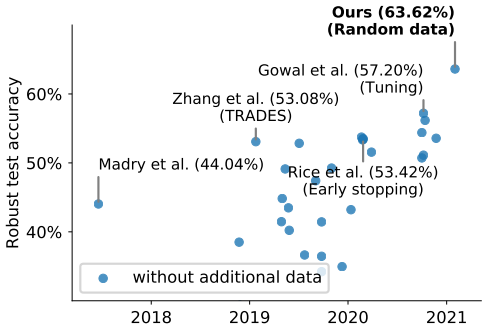


Figure 1: Robust accuracy of models against AUTOATTACK (Croce & Hein, 2020) on CIFAR-10 with ℓ_∞ perturbations of size $8/255$ displayed in publication order. Our method explores how generated data can be used to improve robust accuracy by +6.42% without using any additional external data. This constitutes the largest jump in robust accuracy in this setting in the past 2 years. Our best model reaches a robust accuracy of 63.58% against AA+MT (Gowal et al., 2020).

The adversarial training procedure proposed by Madry et al. (2018) feeds adversarially perturbed examples back into the training data. It is widely regarded as one of the most successful method to train robust deep neural networks (Gowal et al., 2020), and it has been augmented in different ways – with changes in the attack procedure (Dong et al., 2018), loss function (Mosbach et al., 2018; Zhang

et al., 2019) or model architecture (Xie et al., 2019; Zoran et al., 2019). We highlight the works by Carmon et al. (2019); Uesato et al. (2019); Najafi et al. (2019); Zhai et al. (2019) who simultaneously proposed the use of additional external data. While the addition of external data helped boost robust accuracy by a large margin, progress in the setting without additional data has slowed (see Fig. 1). On CIFAR-10 against ℓ_∞ perturbations of size $\epsilon = 8/255$, the best known model obtains a robust accuracy of 65.87% when using additional data. The same model obtains a robust accuracy of 57.14% without this data (Gowal et al., 2020). As a result, we ask ourselves whether it is possible to leverage the information contained in the original training set to a greater extent. This manuscript challenges the status-quo, where it is widely believed that generative models lack diversity and that the samples they produce cannot be used to train classifiers to the same accuracy than those trained on original datasets (Ravuri & Vinyals, 2019). We make the following contributions:

- We demonstrate that it is possible to use low-quality random inputs (sampled from a conditional Gaussian fit of the training data) to improve robust accuracy on CIFAR-10 against ℓ_∞ perturbations of size $\epsilon = 8/255$ (+0.93% on a WRN-28-10) and provide justification in App. G.
- We leverage higher quality generated inputs (i.e., inputs generated by generative models solely trained on the original data), and study three recent generative models: the Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2021), the Very Deep Variational Auto-Encoder (VDVAE) (Child, 2021) and BigGAN (Brock et al., 2018).
- We show that images generated by the DDPM allow us to reach a robust accuracy of 63.58% on CIFAR-10 against ℓ_∞ perturbations of size $\epsilon = 8/255$ (an improvement of +6.44% upon the state-of-the-art). Notably, our best CIFAR-10 and CIFAR-100 models beat all techniques that use additional data, except for the work by Gowal et al. (2020).

2 METHOD

Motivation. Data augmentation has been shown to reduce the generalization error of standard (non-robust) training. However, to the contrary of standard training, augmentations beyond random flips, rotations and crops (He et al., 2016) – such as *Cutout* (DeVries & Taylor, 2017), *mixup* (Zhang et al., 2018a), *AutoAugment* (Cubuk et al., 2019) or *RandAugment* (Cubuk et al., 2020) – have been unsuccessful in the context of adversarial training (Rice et al., 2020; Gowal et al., 2020; Wu et al., 2020). The gap in robust accuracy between models trained with and without additional data suggests that common augmentation techniques, which tend to produce augmented views that are close to the original image they augment, are intrinsically limited in their ability to improve robust generalization. This phenomenon is particularly exacerbated when training adversarially robust models which are known to require an amount of data polynomial in the number of input dimensions (Schmidt et al., 2018). The appendix contains a more complete section on related work.

Hypothesis. We hypothesize that, to improve robust generalization, it is critical to create augmentations that are more diverse and complement the training set. To test our hypothesis, we propose to use samples generated from a simple class-conditional Gaussian fit of the training data. By construction, such samples (shown in App. D) are extremely blurry but diverse. We proceed by fitting a multivariate Gaussian to each set of 5K training images corresponding to each class in CIFAR-10. For each class, we sample 100K images resulting in a new dataset of 1M datapoints. In Fig. 2, we train various robust models by decreasing the ratio of original-to-generated samples present in each batch from 100% (original data only) to 0% (generated data only). Decreasing this ratio reduces the importance of the original data. We observe that all ratios between 50% and 90% provide improvements in robust accuracy. Most surprisingly, the optimal ratio of 80% provides

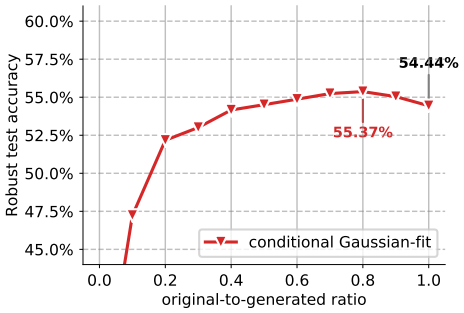


Figure 2: Robust test accuracy (against AA+MT) when training a Wide ResNet (WRN)-28-10 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ on CIFAR-10 when using additional data randomly sampled from a class-conditional Gaussian fit of the training data. We compare how the ratio between original CIFAR-10 images and generated images in the training minibatches affects the test robust performance (0 means generated samples only, while 1 means original CIFAR-10 train set only).

Setup	Neighbor Distribution				Inception Metrics	
	Train	Test	Self	Entropy ↑	Is ↑	Fid ↓
Extracted from 80M-Ti (Carmon et al., 2019)	30.12%	29.20%	40.68%	1.09	11.78 ± 0.12	2.80
mixup (Zhang et al., 2018a)	95.93%	0.32%	3.75%	0.18	9.33 ± 0.22	7.71
Cutout (DeVries & Taylor, 2017)	94.15%	0.22%	5.63%	0.23	8.42 ± 0.16	21.05
Class-conditional Gaussian-fit	0.73%	0.72%	98.55%	0.09	3.64 ± 0.03	117.62
VDVAE (Child, 2021)	6.71 %	5.76%	87.53%	0.46	6.88 ± 0.05	26.44
BigGAN (Brock et al., 2018)	11.53%	10.51%	77.96%	0.68	**9.73**± 0.07	13.78
DDPM (Ho et al., 2021)	21.79%	20.16%	58.05%	**0.97**	9.41 ± 0.13	**6.84**

Table 1: We sample 10K images from common data augmentations applied to the train set and from different generative models. For each sample in each augmented set, we find its closest neighbor in LPIPS (Zhang et al., 2018b) feature space. We report the proportion of samples with a nearest neighbor in either the train set, test set or the sampled set itself (we do not match a sample with itself). We also report the entropy (computed with the natural logarithm) of the nearest neighbor proportions (higher is better), and include the Inception Score (IS) and Frechet Inception Distance (FID) computed from 50K samples from each set. More information on how exactly this table is computed and how samples are generated is available in the appendix.

an absolute improvement of +0.93%, which is an improvement comparable in size to the ones provided by model weight averaging or TRADES (Gowal et al., 2020). See App. G for a theoretical justification that explains why seemingly random data can help improve robustness.

Generative models. This discovery strongly suggests that generative models, which are capable of creating novel images, are viable augmentation candidates (OpenAI, 2021). In this work, we limit ourselves to generative models that are solely trained on the original train set, as we focus on how to improve robustness in the setting without external data. We consider three recent and fundamentally different models: *(i)* BigGAN (Brock et al., 2018): one of the first large-scale application of Generative Adversarial Networks (GANs) which produced significant improvements in Frechet Inception Distance (FID) and Inception Score (IS) on CIFAR-10 (as well as on IMAGENET); *(ii)* VDVAE (Child, 2021): a hierarchical Variational AutoEncoder (VAE) which outperforms alternative VAE baselines; and *(iii)* DDPM (Ho et al., 2021): a diffusion probabilistic model based on Langevin dynamics that reaches state-of-the-art FID on CIFAR-10.[1] As we have done for the simpler class-conditional Gaussian-fit, for each model, we sample 100K images per class, resulting in 1M images in total (the exact procedure is explained in the appendix). A few samples are shown in App. D.

Comparison with common augmentations. In Table 1, we sub-sample 10K images from each generated dataset (corresponding to each generative model). We also apply two common data augmentation techniques (i.e., *mixup* and *Cutout*) to 10K images from the CIFAR-10 train set.[2] For each augmented or generated sample, we find its closest neighbor in LPIPS (Zhang et al., 2018b) feature space (more details are available in the appendix). An ideal generative model, exemplified by samples extracted from 80M-TI, should create samples that are equally likely to be close to images from the train set, from the test set or from the generated set itself. We observe that common augmentation techniques tend to produce samples that are too close to the original train set and that lack complementarity, potentially explaining their limited usefulness in terms of improving adversarial robustness. Meanwhile, generated samples (including those from the class-conditional Gaussian-fit) are much more likely to be close to images of the test set. We also observe that the DDPM neighbor distribution matches more closely the distribution from real, non-generated images extracted from 80M-TI. Images generated by BigGAN and VDVAE tend to have their nearest neighbor among themselves which indicates that these samples are either far from the train and test distributions or produce overly similar samples. For completeness, we also report IS and FID metrics.

3 Experimental results

The full experimental setup is explained in App. B. Specifically, we use WRNs (He et al., 2016; Zagoruyko & Komodakis, 2016) with Swish/SiLU (Hendrycks & Gimpel, 2016) activation functions. We use stochastic weight averaging (Izmailov et al., 2018) with a decay rate of $\tau = 0.995$. For adversarial training, we use TRADES (Zhang et al., 2019) with 10 Projected Gradient Descent (PGD) steps. We train for 800 CIFAR-10-equivalent epochs with a batch size of 1024. As a comparison

[1] For VDVAE and DDPM, we use CIFAR-10 checkpoints available online (we confirmed with their authors that these checkpoints were solely trained on the CIFAR-10 train set). For BigGAN, we trained our own model and matched its IS to the one obtained by DDPM. More details are in the appendix.

[2] According to prior work, both techniques are unable to improve boost robust accuracy beyond the one obtained with standard random cropping/flipping when using early stopping (Rice et al., 2020).

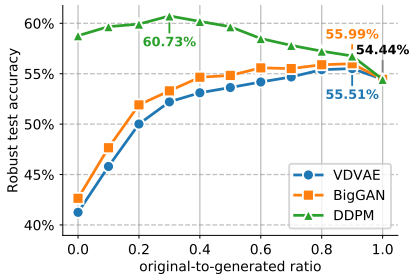


Figure 3: Robust test accuracy (against AA+MT; Gowal et al., 2020) when training a WRN-28-10 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ on CIFAR-10 when using additional data produced by different generative models. We compare how the ratio between original images and generated images in the training minibatches affects the test robust performance (0 means generated samples only, while 1 means original CIFAR-10 train set only).

MODEL	DATASET	NORM	CLEAN	ROBUST
Wu et al. (2020) (WRN-34-10)	CIFAR-10	ℓ_∞	85.36%	56.17%
Gowal et al. (2020) (WRN-70-16)			85.29%	57.14%
Ours (DDPM) (WRN-28-10)			85.97%	60.73%
Ours (DDPM) (WRN-70-16)			86.94%	**63.58%**
Wu et al. (2020) (WRN-34-10)	CIFAR-10	ℓ_2	88.51%	73.66%
Gowal et al. (2020) (WRN-70-16)			90.90%	74.50%
Ours (DDPM) (WRN-28-10)			90.24%	77.37%
Ours (DDPM) (WRN-70-16)			90.83%	**78.31%**
Cui et al. (2020) (WRN-34-10)	CIFAR-100	ℓ_∞	60.64%	29.33%
Gowal et al. (2020) (WRN-70-16)			60.86%	30.03%
Ours (DDPM) (WRN-28-10)			59.18%	30.81%
Ours (DDPM) (WRN-70-16)			60.46%	**33.49%**
Ours (without DDPM) (WRN-28-10)	SVHN	ℓ_∞	92.87%	56.83%
Ours (DDPM) (WRN-28-10)			94.15%	**60.90%**

Table 2: Clean (without perturbations) and robust (under adversarial attack) accuracy obtained by different models (we pick the worst accuracy obtained by either AUTOATTACK or AA+MT). The accuracies are reported on the full test sets. For CIFAR-10, we test against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ and ℓ_2 norm-bounded perturbations of size $\epsilon = 128/255$. For CIFAR-100 and SVHN, we test against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$.

point, we trained ten WRN-28-10 models on CIFAR-10. The resulting robust test accuracy on CIFAR-10 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ is 54.44±0.39%, thus showing a relatively low variance in the results. Furthermore, as we will see, our best models are well clear of the threshold for statistical significance.

Mixing ratio. As done for the class-conditional Gaussian-fit in Sec. 2, we vary the ratio original-to-generated images in each mini-batch for all three generated datasets. Fig. 3 explores a wide range of ratios while training a WRN-28-10 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ on CIFAR-10. A ratio of zero indicates that only generated images are used, while a ratio of one indicates that only images from the CIFAR-10 train set are used. Samples from all models improve robustness when mixed optimally, but only samples from the DDPM improve robustness significantly. It is also interesting to observe that, in this case, using 1M generated images is better than using the 50K images from the original train set only. While this may seem surprising, it can easily be explained if we assume that the DDPM produces many more high-quality, high-diversity images than the limited set of images present in the original data (c.f., Schmidt et al., 2018). Overall, DDPM samples significantly boosts the robust accuracy with an improvement of +6.29% compared to using the original train set only, whereas using BigGAN and VDVAE samples result in smaller (although significant) improvements upon the baseline with +1.55% and +1.07%, respectively.

CIFAR-10. Table 2 shows the performance of models trained with samples generated by the DDPM on CIFAR-10 against ℓ_∞ and ℓ_2 norm-bounded perturbations of size $\epsilon = 8/255$ and $\epsilon = 128/255$, respectively. Irrespective of their size, models trained with DDPM samples surpass the current state-of-the-art in robust accuracy by a large margin (+6.44% and +3.81%). Most remarkably, we highlight that, despite not using any external data, our best models beat all RobustBench (https://robustbench.github.io/) entries that used external data (except for one).

CIFAR-100 and SVHN. Finally, to evaluate the generality of our approach, we evaluate our approach on CIFAR-100. We train a new DDPM ourselves on the train set of CIFAR-100 and sample 1M images. The results are shown in Table 2. Our best model reaches a robust accuracy of 33.49% and improves noticeably on the state-of-the-art (in the setting that does not use any external data). On SVHN, in the same table, we compare models trained without and with DDPM samples. Again, the addition of DDPM samples significantly improves robustness.

4 CONCLUSION

Using generative models, we posit and demonstrate that generated samples provide a greater diversity of augmentations that allow adversarial training to go well beyond the state-of-the-art. Our work provides novel insights into the effect of diversity and complementarity on robustness, which we hope can further our understanding of robustness. All our models and generated datasets are available online at https://github.com/deepmind/deepmind-research/tree/master/adversarial_robustness/iclrw2021doing.

REFERENCES

Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *Int. Conf. Mach. Learn.*, 2018.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *Int. Conf. Mach. Learn.*, 2018.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *NIPS Deep Learning Symposium*, 2016.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *Int. Conf. Learn. Represent.*, 2018.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017a.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, 2017b.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Adv. Neural Inform. Process. Syst.*, 2019.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *Int. Conf. Learn. Represent.*, 2021.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *arXiv preprint arXiv:2003.01690*, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. *arXiv preprint arXiv:2011.11164*, 2020. URL https://arxiv.org/pdf/2011.11164.

Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, George van den Driessche, Balaji Lakshminarayanan, Clemens Meyer, Faith Mackinder, Simon Bouton, Kareem Ayoub, Reena Chopra, Dominic King, Alan Karthikesalingam, Cían O Hughes, Rosalind Raine, Julian Hughes, Dawn A Sim, Catherine Egan, Adnan Tufail, Hugh Montgomery, Demis Hassabis, Geraint Rees, Trevor Back, Peng T Khaw, Mustafa Suleyman, Julien Cornebise, Pearse A Keane, and Olaf Ronneberger. Clinically applicable deep learning for diagnosis and referral in retinal disease. In *Nature Medicine*, 2018.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting Adversarial Attacks with Momentum. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *Int. Conf. Learn. Represent.*, 2015.

Sven Gowal, Chongli Qin, Po-Sen Huang, Taylan Cemgil, Krishnamurthy Dvijotham, Timothy Mann, and Pushmeet Kohli. Achieving Robustness in the Wild via Adversarial Mixing with Disentangled Representations. *arXiv preprint arXiv:1912.03192*, 2019a. URL https://arxiv.org/pdf/1912.03192.

Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy Mann, and Pushmeet Kohli. An Alternative Surrogate Loss for PGD-based Adversarial Testing. *arXiv preprint arXiv:1910.09338*, 2019b.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. URL https://arxiv.org/pdf/2010.03593.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/pdf?id=HJz6tiCqYm.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. *arXiv preprint arXiv:2006.16241*, 2020. URL https://arxiv.org/pdf/2006.16241.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.*, 2021.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering Interpretable GAN Controls. *arXiv preprint arXiv:2004.02546*, 2020. URL https://arxiv.org/pdf/2004.02546.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging Weights Leads to Wider Optima and Better Generalization. *Uncertainty in Artificial Intelligence*, 2018.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/pdf?id=HylsTT4FvB.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR workshop*, 2016.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *Int. Conf. Learn. Represent.*, 2017.

Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Learning to generate noise for robustness against multiple perturbations. *arXiv preprint arXiv:2006.12135*, 2020. URL https://arxiv.org/pdf/2006.12135.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *Int. Conf. Learn. Represent.*, 2018.

Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit Pairing Methods Can Fool Gradient-Based Attacks. *arXiv preprint arXiv:1810.12042*, 2018.

Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. *Adv. Neural Inform. Process. Syst.*, 2019.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Sov. Math. Dokl*, 1983.

OpenAI. Dall-e: Creating images from text, 2021. URL https://openai.com/blog/dall-e.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020a.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Hang Su, and Jun Zhu. Boosting Adversarial Training with Hypersphere Embedding. *Adv. Neural Inform. Process. Syst.*, 2020b.

Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=H1laeJrKDB.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 1964.

Suman Ravuri and Oriol Vinyals. Classification Accuracy Score for Conditional Generative Models. *arXiv preprint arXiv:1905.10887*, 2019. URL https://arxiv.org/pdf/1905.10887.pdf.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. *Int. Conf. Mach. Learn.*, 2020.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially Robust Generalization Requires More Data. *Adv. Neural Inform. Process. Syst.*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *Int. Conf. Learn. Represent.*, 2014.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble Adversarial Training: Attacks and Defenses. *arXiv preprint arXiv:1705.07204*, 2017. URL https://arxiv.org/pdf/1705.07204.

Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. *Int. Conf. Mach. Learn.*, 2018.

Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *Adv. Neural Inform. Process. Syst.*, 2019.

Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf?id=MIDckA56aD.

Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu. Do wider neural networks really help adversarial robustness? *arXiv preprint arXiv:2010.01279*, 2021. URL https://arxiv.org/pdf/2010.01279.

Dongxian Wu, Shu-tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Adv. Neural Inform. Process. Syst.*, 2020.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *Brit. Mach. Vis. Conf.*, 2016.

Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially Robust Generalization Just Requires More Unlabeled Data. *arXiv preprint arXiv:1906.00555*, 2019.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. *Int. Conf. Mach. Learn.*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *Int. Conf. Learn. Represent.*, 2018a.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint arXiv:1801.03924*, 2018b. URL https://arxiv.org/pdf/1801.03924.

Daniel Zoran, Mike Chrzanowski, Po-Sen Huang, Sven Gowal, Alex Mott, and Pushmeet Kohl. Towards Robust Image Classification Using Sequential Attention Models. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Doing More with Less: Improving Robustness using Generated Data (Supplementary Material)

A Related work

Adversarial ℓ_p-norm attacks. Since Szegedy et al. (2014) observed that neural networks which achieve high accuracy on test data are highly vulnerable to adversarial examples, the art of crafting increasingly sophisticated adversarial examples has received a lot of attention. Goodfellow et al. (2015) proposed the Fast Gradient Sign Method (FGSM) which generates adversarial examples with a single normalized gradient step. It was followed by R+FGSM (Tramèr et al., 2017), which adds a randomization step, and the Basic Iterative Method (BIM) (Kurakin et al., 2016), which takes multiple smaller gradient steps.

Adversarial training as a defense. The adversarial training (Madry et al., 2018) is widely regarded as one of the most successful method to train robust deep neural networks. It has received significant attention and various modifications have emerged (Dong et al., 2018; Mosbach et al., 2018; Xie et al., 2019). A notable work is TRADES (Zhang et al., 2019), which balances the trade-off between standard and robust accuracy, and achieved state-of-the-art performance against ℓ_∞ norm-bounded perturbations on CIFAR-10. More recently, the work from Rice et al. (2020) studied *robust overfitting* and demonstrated that improvements similar to TRADES could be obtained more easily using classical adversarial training with early stopping. Finally, Gowal et al. (2020) highlighted how different hyper-parameters (such as network size and model weight averaging) affect robustness.

Data-driven data augmentation. Work, such as *AutoAugment* (Cubuk et al., 2019) and related *RandAugment* (Cubuk et al., 2020), learn augmentation policies directly from data. These methods are tuned to improve standard classification accuracy and have been shown to work well on CIFAR-10, CIFAR-100, SVHN and IMAGENET. DeepAugment (Hendrycks et al., 2020) explores how perturbations of the parameters of several image-to-image models can be used to generate augmented datasets that provide increased robustness to common corruptions (Hendrycks & Dietterich, 2018). Similarly, generative models can be used to create novel views of images (Plumerault et al., 2020; Jahanian et al., 2019; Härkönen et al., 2020) by manipulating them in latent space. When optimized and used during training, these novel views reduce the impact of spurious correlations and improve accuracy (Gowal et al., 2019a; Wong & Kolter, 2021). However, to the best of our knowledge, there is little (Madaan et al., 2020) to no evidence that generative models can be used to improve adversarial robustness against ℓ_p-norm attacks. In fact, generative models mostly lack diversity and it is widely believed that the samples they produce cannot be used to train classifiers to the same accuracy than those trained on original datasets (Ravuri & Vinyals, 2019).

B Experimental setup

Architecture. We use WRNs (He et al., 2016; Zagoruyko & Komodakis, 2016) as our backbone network. This is consistent with prior work (Madry et al., 2018; Rice et al., 2020; Zhang et al., 2019; Uesato et al., 2019; Gowal et al., 2020) which use diverse variants of this network family. Furthermore, we adopt the same architecture details as Gowal et al. (2020) with Swish/SiLU (Hendrycks & Gimpel, 2016) activation functions. Most of the experiments are conducted on a WRN-28-10 model which has a depth of 28, a width multiplier of 10 and contains 36M parameters. To evaluate the effect of data augmentations on wider and deeper networks, we also run several experiments using WRN-70-16, which contains 267M parameters.

Outer minimization. We use TRADES (Zhang et al., 2019) optimized using SGD with Nesterov momentum (Polyak, 1964; Nesterov, 1983) and a global weight decay of 5×10^{-4}. When using additional generated data, we increase the batch size to 1024 with a ratio of original-to-added data of 0.3 (unless stated otherwise), train for 800 CIFAR-10-equivalent epochs, and use a *cosine* learning rate schedule (Loshchilov & Hutter, 2017) without restarts where the initial learning rate is set to 0.1 and is decayed to 0 by the end of training (similar to Gowal et al., 2020). We scale the learning rates using

the linear scaling rule of Goyal et al. (2017) (i.e., effective LR $= \max(\text{LR} \times \text{batch size}/256, \text{LR})$). We also use model weight averaging (WA) (Izmailov et al., 2018). The decay rate of WA is set to $\tau = 0.995$. Finally, to use additional generated data with TRADES, we annotate the extra data with the pseudo-labeling technique described by Carmon et al. (2019) where a separate classifier trained on clean CIFAR-10 data provides labels to the unlabeled samples.

Inner minimization. Adversarial examples are obtained by maximizing the Kullback-Leibler divergence between the predictions made on clean inputs and those made on adversarial inputs (Zhang et al., 2019). This optimization procedure is done using the standard PGD formulation (Kurakin et al., 2016) with step-size $\epsilon/4$ and 10 steps.

Evaluation. We follow the evaluation protocol designed by Gowal et al. (2020). Specifically, we train two (and only two) models for each hyperparameter setting, perform early stopping for each model on a separate validation set of 1024 samples using PGD^{40} similarly to Rice et al. (2020) and pick the best model by evaluating the robust accuracy on the same validation set . Finally, we report the robust test accuracy against a mixture of AUTOATTACK (Croce & Hein, 2020) and MULTITARGETED (Gowal et al., 2019b), which is denoted by AA+MT. This mixture consists in completing the following sequence of attacks: AUTOPGD on the cross-entropy loss with 5 restarts and 100 steps, AUTOPGD on the difference of logits ratio loss with 5 restarts and 100 steps and finally MULTITARGETED on the margin loss with 10 restarts and 200 steps.

C ANALYSIS OF MODELS

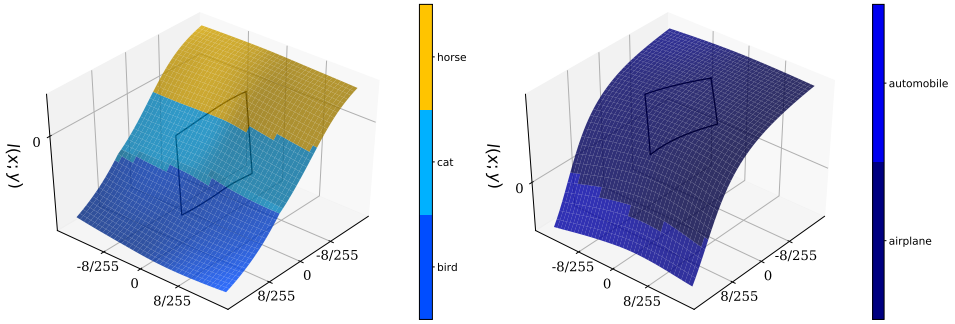
In this section, we perform additional diagnostics that give us confidence that our models are not doing any form of gradient obfuscation or masking (Athalye et al., 2018; Uesato et al., 2018).

AUTOATTACK and robustness against black-box attacks. First, we report in Table 3 the robust accuracy obtained by our strongest models against a diverse set of attacks. These attacks are run as a cascade using the AUTOATTACK library available at https://github.com/fra31/auto-attack. First, we observe that our combination of attacks, denoted AA+MT matches the final robust accuracy measured by AUTOATTACK. Second, we also notice that the black-box attack (i.e., SQUARE) does not find any additional adversarial examples. Overall, these results indicate that our empirical measurement of robustness is meaningful and that our models do not obfuscate gradients.

MODEL	DATASET	NORM	RADIUS	AUTOPGD-CE	+ AUTOPGD-T	+ FAB-T	+ SQUARE	CLEAN	AA+MT
WRN-28-10 (DDPM)	CIFAR-10	ℓ_∞	$\epsilon = 8/255$	63.53%	60.73%	60.73%	60.73%	85.97%	60.73%
WRN-70-16 (DDPM)				65.95%	63.62%	63.62%	63.62%	86.94%	63.58%
WRN-28-10 (DDPM)	CIFAR-10	ℓ_2	$\epsilon = 128/255$	78.13%	77.44%	77.44%	77.44%	90.24%	77.37%
WRN-70-16 (DDPM)				78.97%	78.39%	78.39%	78.39%	90.93%	78.31%
WRN-28-10 (DDPM)	CIFAR-100	ℓ_∞	$\epsilon = 8/255$	34.47%	30.81%	30.81%	30.81%	59.18%	31.23%
WRN-70-16 (DDPM)				36.27%	33.49%	33.49%	33.49%	60.46%	33.93%

Table 3: Clean (without adversarial attacks) accuracy and robust accuracy (against the different stages of AUTOATTACK) on CIFAR-10 obtained by different models. Refer to https://github.com/fra31/auto-attack for more details.

Loss landscapes. We analyze the adversarial loss landscapes of our best model trained on CIFAR-10 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ (a WRN-70-16). To generate a loss landscape, we vary the network input along the linear space defined by the worse perturbation found by PGD^{40} (u direction) and a random Rademacher direction (v direction). The u and v axes represent the magnitude of the perturbation added in each of these directions respectively and the z axis is the adversarial margin loss (Carlini & Wagner, 2017b): $z_y - \max_{i \neq y} z_i$ (i.e., a misclassification occurs when this value falls below zero). Fig. 4 shows the loss landscapes around the first 2 images of the CIFAR-10 test set for the aforementioned model. Both landscapes are smooth and do not exhibit patterns of gradient obfuscation. Overall, it is difficult to interpret these figures further, but they do complement the numerical analyses done so far.



(a) Image of a horse (b) Image of an airplane

Figure 4: Loss landscapes around the first two images from the CIFAR-10 test set for the WRN-70-16 networks trained with DDPM samples. It is generated by varying the input to the model, starting from the original input image toward either the worst attack found using PGD^{40} (u direction) or a random Rademacher direction (v direction). The loss used for these plots is the margin loss $z_y - \max_{i \neq y} z_i$ (i.e., a misclassification occurs when this value falls below zero). The diamond-shape represents the projected ℓ_∞ ball of size $\epsilon = 8/255$ around the nominal image.

D DETAILS ON GENERATED DATA

Generative models. In this paper, we use three different and complementary generative models: *(i)* BigGAN (Brock et al., 2018): one of the first large-scale application of GANs which produced significant improvements in FID and IS on CIFAR-10 (as well as on IMAGENET); *(ii)* VDVAE Child (2021): a hierarchical VAE which outperforms alternative VAE baselines; and *(iii)* DDPM Ho et al. (2021): a diffusion probabilistic model based on Langevin dynamics that reaches state-of-the-art FID on CIFAR-10. Except for BigGAN, we use the CIFAR-10 checkpoints that are available online. For BigGAN, we train our own model and pick the model that achieves the best FID (the model architecture and training schedule is the same as the one used by Brock et al., 2018). All models are trained solely on the CIFAR-10 train set. As a baseline, we also fit a class-conditional multivariate Gaussian, which reaches FID and IS metrics of 120.63 and 3.49, respectively. We also report that BigGAN reaches an FID of 11.07 and IS of 9.71; VDVAE reaches an FID of 36.88 and IS of 6.03; and DDPM reaches an FID of 3.28 and IS of 9.44.[3,4]

Datasets of generated samples. We sample from each generative model 5M images. Similarly to Carmon et al. (2019), we score each image using a pretrained WRN-28-10. This WRN-28-10 is trained non-robustly on the CIFAR-10 train set and achieves 95.68% accuracy.[5,6] For each class, we select the top-100K scoring images and build a dataset of 1M image-label pairs.[7] This additional generated data is used to train adversarially robust models by mixing for each minibatch a given proportion of original and generated examples. Fig. 5 shows a random subset of this additional data for each generative model. We also report the FID and IS metrics of the resulting sets in Table 1. They differ from the metrics obtained by each generative model as we filter images to only pick the highest scoring ones.

Diversity and complementarity. While the FID metric does capture how two distributions of samples match, it does not necessarily provide enough information in itself to assess the overlap between the distribution of generated samples and the train or test distributions (this is especially true for samples obtained through common data augmentations). As such, we also decide to compute the proportion of nearest neighbors in perceptual space. Given equal Inception metrics, a better

[3]For CIFAR-100, we trained our own DDPM which achieves an FID of 5.58 and IS of 10.82.

[4]For SVHN, we trained our own DDPM which achieves an FID of 4.89 and IS of 3.06.

[5]For CIFAR-100, the same model achieves 79.98% accuracy.

[6]For SVHN, the same model achieves 96.54% accuracy.

[7]All generated datasets are available online at https://github.com/deepmind/deepmind-research/tree/master/adversarial_robustness/iclrw2021doing.

generative model would produce samples that are equally likely to be close to training, testing or generated images.

We now describe how we compute Table 1 which reports the nearest-neighbors statistics for the different augmentation methods. First, we sample 10K images from the train set of CIFAR-10 (uniformly across classes) and take the full test set of CIFAR-10. We then pass these 20K images through the pretrained VGG network which measures a Perceptual Image Patch Similarity, also known as LPIPS (Zhang et al., 2018b). We use the resulting concatenated activations and compute their top-100 PCA components, as this allows us to compare samples in a much lower dimensional space (i.e., 100 instead of 124,928). Finally, for each augmentation method (heuristics- or data-driven), we sample 10K images and pass them through the pipeline composed of the LPIPS VGG network and the PCA projection computed on the original data. For each sample, we find its closest neighbor in the PCA-reduced feature space and measure whether this nearest-neighbor belongs to the original dataset (train or test) or to the generated set (self) of 10K images.

(a) Conditional Gaussian (b) VDVAE

(c) BigGAN (d) DDPM

Figure 5: CIFAR-10 samples generated by different approaches and used as additional data to train adversarially robust models. Each row correspond to a different class in the following order: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck. Each image is assigned a *pseudo-label* using a standard classifier trained on the CIFAR-10 train set.

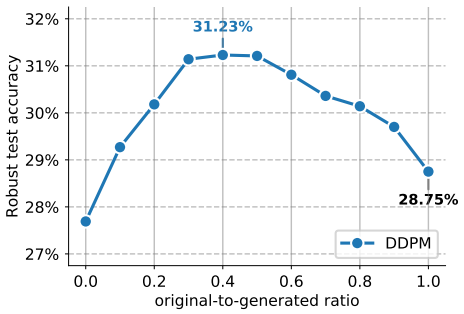


Figure 6: Robust test accuracy (against AA+MT) when training a WRN-28-10 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ on CIFAR-100 when using additional data produced by a DDPM. We compare how the ratio between original images and generated images in the training minibatches affects the test robust performance (0 means generated samples only, while 1 means original CIFAR-100 train set only).

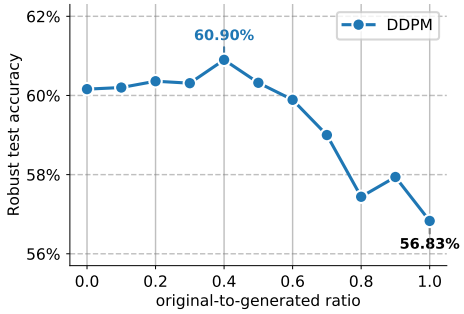


Figure 7: Robust test accuracy (against AA+MT) when training a WRN-28-10 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ on SVHN when using additional data produced by a DDPM. We compare how the ratio between original images and generated images in the training minibatches affects the test robust performance (0 means generated samples only, while 1 means original SVHN train set only).

E ADDITIONAL RESULTS

CIFAR-100. For completeness, we also report the effect of mixing different proportions of generated and original samples in Fig. 6 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ using a WRN-28-10 on CIFAR-100. Similarly to Fig. 3, we observe that additional samples generated by DDPM are useful to improve robustness, with an absolute improvement of +2.48% in robust accuracy.

SVHN. We report the effect of mixing different proportions of generated and original samples in Fig. 6 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$ using a WRN-28-10 on SVHN. Similarly to Fig. 3 and Fig. 6, we observe that additional samples generated by DDPM are useful to improve robustness, with an absolute improvement of +4.07% in robust accuracy.

F ROBUSTBENCH

For reference, at the time of writing, the top-5 RobustBench (https://robustbench.github.io/; Croce et al., 2020) leaderboard entries without and with additional data are listed in Table 4.

AUTHOR	MODEL	CLEAN	ROBUST
WITHOUT EXTERNAL DATA			
Gowal et al. (2020)	WRN-70-16	85.29%	57.14%
Gowal et al. (2020)	WRN-34-20	85.64%	56.82%
Wu et al. (2020)	WRN-34-10	85.36%	56.17%
Pang et al. (2020a)	WRN-34-20	86.43%	54.39%
Pang et al. (2020b)	WRN-34-20	85.14%	53.74%
WITH EXTERNAL DATA			
Gowal et al. (2020)	WRN-70-16	91.10%	65.87%
Gowal et al. (2020)	WRN-34-20	89.48%	62.76%
Wu et al. (2021)	WRN-34-15	87.67%	60.65%
Wu et al. (2020)	WRN-28-10	88.25%	60.04%
Carmon et al. (2019)	WRN-28-10	89.69%	59.53%

Table 4: State of RobustBench leaderboard at the time of writing. We report the clean (without adversarial attacks) accuracy and robust accuracy on CIFAR-10 against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$.

G RANDOMNESS IS ENOUGH

In this section, we provide three sufficient conditions that explain why generated data helps improve robustness: *(i)* the pre-trained, non-robust classifier used for pseudo-labeling (see App. D) must be accurate enough, *(ii)* the likelihood of sampling examples that are adversarial to this non-robust classifier must be low, and *(iii)* it is possible to sample *real* images with enough frequency.

G.1 SETUP

Given access to a pre-trained non-robust classifier $f_{\text{NR}} : \mathcal{X}_{\text{all}} \mapsto \mathcal{Y}$ and an unconditional generative model approximating the true data distribution $p^\star$ by a distribution $\hat{p}$ over $\mathcal{X}_{\text{all}}$, we would like to train a robust classifier $f_{\text{R}}^{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta}$.

The set of inputs $\mathcal{X}_{\text{all}} = \{0, 1/255, \dots, 1\}^n$ is the set of all images (discretized and scaled between 0 and 1) with dimensionality n.[8] The set of labels $\mathcal{Y} \in 2^{\mathbb{Z}}$ is a set of integers, each of which represents a given class (e.g., *dog* as opposed to *cat*). There exists an image manifold $\mathcal{X} \subseteq \mathcal{X}_{\text{all}}$ that contains all *real* images (i.e., images for which we want to enforce robustness). The distribution of *real* images is denoted $p^\star$ with $\mathbb{P}_{\boldsymbol{x}\sim p^\star}(\boldsymbol{x}) > 0$ if $\boldsymbol{x} \in \mathcal{X}$ and $\mathbb{P}_{\boldsymbol{x}\sim p^\star}(\boldsymbol{x}) = 0$ otherwise. We further assume that each image $\boldsymbol{x} \in \mathcal{X}$ can be assigned one and only one label $y = f^\star(\boldsymbol{x})$ where $f^\star : \mathcal{X} \mapsto \mathcal{Y}$ is a perfect classifier (only valid for *real* images). Given a perturbation ball $\mathbb{S}(\boldsymbol{x})$[9], we restrict labels such that there exists no *real* image within the perturbation ball of another that has a different label; i.e., $\forall \boldsymbol{x}' \in \mathbb{S}(\boldsymbol{x}) \cap \mathcal{X}$ we have $f^\star(\boldsymbol{x}) = f^\star(\boldsymbol{x}')$ for all $\boldsymbol{x} \in \mathcal{X}$.

Overall, we would like find optimal parameters $\boldsymbol{\theta}^\star$ for $f_{\text{R}}^{\boldsymbol{\theta}^\star}$ that minimize the adversarial risk,

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim p^\star} \left[\max_{\boldsymbol{x}' \in \mathbb{S}(\boldsymbol{x})} \mathbb{1}_{f_{\text{R}}^{\boldsymbol{\theta}}(\boldsymbol{x}') \neq f^\star(\boldsymbol{x})} \right] \tag{1}$$

without enumerating all *real* images or the ideal classifier. As such, we settle for the following sub-optimal parameters

$$\hat{\boldsymbol{\theta}}^\star = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim \hat{p}} \left[\max_{\boldsymbol{x}' \in \mathbb{S}(\boldsymbol{x})} \mathbb{1}_{f_{\text{R}}^{\boldsymbol{\theta}}(\boldsymbol{x}') \neq f_{\text{NR}}(\boldsymbol{x})} \right]. \tag{2}$$

Relationship to our method. The above setting corresponds to the one studied in the main manuscript where a generative model is trained on a limited number of samples from the true data distribution $\mathbb{D}_{\text{train}} = \{x_i \sim p^\star\}_{i=1}^N$. During training, we mix *real* and *generated* samples and solve the following problem:

$$\arg\min_{\boldsymbol{\theta}} \alpha \cdot \mathbb{E}_{\boldsymbol{x}\in\mathbb{D}} \left[\max_{\boldsymbol{x}' \in \mathbb{S}(\boldsymbol{x})} l_{\text{ce}} \left(f_{\text{R}}^{\boldsymbol{\theta}}(\boldsymbol{x}'), f^\star(\boldsymbol{x}) \right) \right] + (1-\alpha) \cdot \mathbb{E}_{\boldsymbol{x}\sim\hat{p}'} \left[\max_{\boldsymbol{x}' \in \mathbb{S}(\boldsymbol{x})} l_{\text{ce}} \left(f_{\text{R}}^{\boldsymbol{\theta}}(\boldsymbol{x}'), f'_{\text{NR}}(\boldsymbol{x}) \right) \right]. \tag{3}$$

where α is the ratio of original-to-generated samples (see Sec. 2), f'_{NR} is the underlying pre-trained classifier (used for generated samples only), $\hat{p}'$ is the generative model distribution (which excludes samples from the train set, e.g. DDPM) and where the 0-1 loss is replaced with the cross-entropy loss l_{ce}. Ignoring the change of loss, Eq. 3 can be formulated exactly as Eq. 2 by having

$$f_{\text{NR}}(x) = \begin{cases} f^\star(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \mathbb{D} \\ f'_{\text{NR}}(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{4}$$

and by sampling a datapoint $\boldsymbol{x}$ from the distribution of our generative model as

$$\boldsymbol{x} = \mathbb{1}_{r\leq\alpha}\boldsymbol{x}' + \mathbb{1}_{r>\alpha}\boldsymbol{x}'' \text{ with } r \sim \mathcal{U}_{[0,1]}, \boldsymbol{x}' \sim \mathcal{U}_{\mathbb{D}} \text{ and } \boldsymbol{x}'' \sim \hat{p}' \tag{5}$$

where $\mathcal{U}_{\mathbb{A}}$ corresponds to the uniform distribution over set $\mathbb{A}$.

G.2 SUFFICIENT CONDITIONS

In order to obtain sub-optimal parameters $\hat{\boldsymbol{\theta}}^\star$ that approach the performance of the optimal parameters $\boldsymbol{\theta}^\star$, the following conditions are sufficient (in the limit of infinite capacity and compute).[10] These provide a deeper understanding of our method.

[8] Discretizing the image space is not necessary, but makes the mathematical notations simpler.

[9] E.g., $\mathbb{S}(\boldsymbol{x}) = \{\boldsymbol{x}' : \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty \leq \epsilon\}$

[10] Understanding to which extent violations of these conditions affect robustness remains part of future work.

Condition 1 (accurate classifier). The pre-trained non-robust classifier f_{NR} must be accurate. When $p^\star = \hat{p}$, Eq. 2 can be made identical to Eq. 1 by setting $f_{\text{NR}}(\boldsymbol{x}) = f^\star(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. For all practical settings, we posit that "good", sub-optimal parameters $\hat{\boldsymbol{\theta}}^\star$ can be obtained even when the non-robust classifier f_{NR} is not perfect. However, it must achieve sufficient accuracy. On CIFAR-10, typical classifiers that are solely trained on images from the train set can reach high accuracy. In our work, we use a pseudo-labeling classifier that achieves 95.68% on the CIFAR-10 test set.

Condition 2 (unlikely attacks). The probability of sampling a point $\boldsymbol{x} \sim \hat{p}$ outside the image manifold such that it is adversarial to f_{NR} is low:

$$\mathbb{P}_{\boldsymbol{x}\sim\hat{p}}\left(\exists \boldsymbol{x}' \in \mathcal{X} \text{ with } f_{\text{NR}}(\boldsymbol{x}) \neq f_{\text{NR}}(\boldsymbol{x}') \text{ and } \boldsymbol{x} \in \mathbb{S}(\boldsymbol{x}')\right) < \delta,\ \delta \geq 0. \tag{6}$$

To understand why this condition is needed, it is worth considering the optimal non-robust classifier $f_{\text{NR}}(\boldsymbol{x}) = f^\star(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. When $p^\star \neq \hat{p}$, it becomes possible to sample points outside the manifold of *real* images and for which no correct labels exist. Fortunately, in the limit of infinite capacity, these points can only influence the accuracy of the final robust classifier on *real* images if they are within an adversarial ball $\mathbb{S}(\boldsymbol{x})$ for a *real* image $\boldsymbol{x}$. In practical settings, it is well documented that random sampling (e.g., using uniform or Gaussian sampling) is unlikely to produce images that are adversarial. Hence, we posit than, unless the generative model represented by $\hat{p}$ is trained to produce adversarial images, this condition is met.

Condition 3 (sufficient coverage). There must a non-zero probability of sampling any point on the image manifold:

$$\mathbb{P}_{\boldsymbol{x}\sim\hat{p}}(\boldsymbol{x}) > 0, \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{7}$$

In other words, the generative model should output a *diverse* set of samples and some of these samples should look like *real* images. Note that it remains possible to obtain a "good", sub-optimal robust classifier when this condition does not hold. However, its accuracy will rapidly decrease as coverage drops. Hence, it is important to avoid using generative models that collapsed to a few modes and exhibit low diversity.

G.3 DISCUSSION

This last condition explains why samples generated by a simple class-conditional Gaussian-fit can be used to improve robustness. Indeed, these conditions imply that it is not necessary to have access to either the true data distribution or a perfect generative model when given enough compute and capacity. However, it is also worth understanding what happens when capacity or compute is limited. In this case, it is critical that the optimization focuses on *real* images and that the distribution $\hat{p}$ be as close as possible to the true distribution $p^\star$. In practice, this translates to the fact that better generative models (such as DDPM) can be used to achieve better robustness.