Improving Robustness using Generated Data

Anonymous Author(s)
Affiliation
Address
email

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 generative models trained solely on the original training set can be leveraged to artificially increase the size of the original training set and improve adversarial robustness to ℓ_p norm-bounded perturbations. We identify the sufficient conditions under which incorporating additional generated data can improve robustness, and demonstrate that it is possible to significantly reduce the robust-accuracy gap to models trained with additional real data. Surprisingly, we show that even the addition of non-realistic random data (generated by Gaussian sampling) can improve robustness. We evaluate our approach on CIFAR-10, CIFAR-100 and SVHN 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 models achieve 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 [12] to autonomous driving [5] via medical diagnostics [19]. 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 [8, 9, 22, 36, 56]. There has been a lot of work on understanding and generating adversarial perturbations [1, 4, 9, 56], and on building defenses that are robust to such perturbations [22, 41, 52, 72]. We note that while robustness and invariance to input perturbations is crucial to the deployment of machine learning models in various applications, it can also have broader negative impacts to society such as hindering privacy [55] or increasing bias [60].

The adversarial training procedure proposed by Madry et al. [41] 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 [25], and it has been augmented in different ways – with changes in the attack procedure [21], loss function [42, 72] or model architecture [67, 75]. We highlight the works by Carmon et al. [10], Najafi et al. [43], Uesato et al. [63], Zhai et al. [70] who simultaneously proposed the use of additional unlabeled 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

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

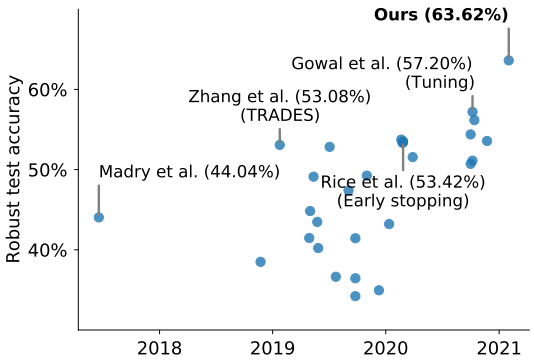


Figure 1: Robust accuracy of models against AUTOATTACK [13] on CIFAR-10 with ℓ_∞ perturbations of size $8/255$ displayed in publication order. Our method explores how generated data can be used to improve robust accuracy by +6.42% without using any additional external data. This constitutes the largest 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 [25].

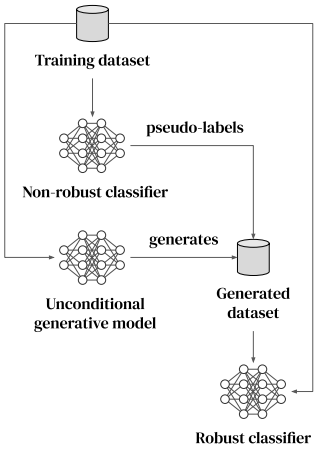


Figure 2: Overview of our approach. Our method initially trains a generative model and a non-robust classifier. The non-robust classifier is used to provide pseudo-labels to the generated data. Finally, generated and original training data are combined to train a robust classifier.

(see Fig. 1). On CIFAR-10 [35] 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 [25]. 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. To the contrary of standard training where it is widely believed that generative models lack diversity and that the samples they produce cannot be used to train better classifiers [51], we demonstrate both theoretically and experimentally that these generated samples can be used to improve robustness (using the approach described in Fig. 2 and Sec. 3.3). We make the following contributions:

- We demonstrate in Sec. 3.2 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 a justification and sufficient conditions in Sec. 4.
- 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) [31], the Very Deep Variational Auto-Encoder (VDVAE) [11] and BigGAN [6] (Sec. 5). We show that DDPM samples cover most closely the real data distribution (as measured by the distance to the test set in the Inception feature space).
- Using images generated by the DDPM allows 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 [35] models beat all techniques that use additional data, except for the work by Gowal et al. [25] (see Sec. 6).

2 Related work

Adversarial ℓ_p norm-bounded attacks. Since Biggio et al. [4], Szegedy et al. [56] 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. [22] proposed the Fast Gradient Sign Method (FGSM) which generates adversarial examples with a single normalized gradient step. It was followed by R+FGSM [59], which adds a randomization step, and the Basic Iterative Method (BIM) [36], which takes multiple smaller gradient steps.

Adversarial training as a defense. Adversarial training [41] is widely regarded as one of the most successful methods to train deep neural networks robust to such attacks. It has received significant

attention and various modifications have emerged [21, 42, 67]. A notable work is TRADES [72], 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. [52] 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. [25] highlighted how different hyper-parameters (such as network size and model weight averaging) affect robustness.

Data-driven augmentations. Works, such as *AutoAugment* [15] and related *RandAugment* [16], learn augmentation policies directly from data. These methods are tuned to improve standard classification accuracy and have been shown to work well on multiple datasets. *DeepAugment* [30] explores how perturbations of the parameters of pre-trained image-to-image models can be used to generate augmented datasets that provide increased robustness to common corruptions [28]. Similarly, generative models can be used to create novel views of images [32, 34, 49] by manipulating them in latent space. When optimized and used during training, these novel views reduce the impact of spurious correlations and improve accuracy [23, 64]. Most recently, Laidlaw et al. [37] proposed an adversarial training method based on bounding a neural perceptual distance (i.e., an approximation of the true perceptual distance). While these works make significant contributions towards improving generalization and robustness to semantic perturbations, they do not improve robustness to ℓ_p norm-bounded perturbations.

Robustness to ℓ_p norm-bounded perturbations using generative modeling. Finally, we highlight works, such as *Defense-GAN* [53] or *ME-Net* [68], which leverage data modeling techniques to create stronger defenses against ℓ_p norm-bounded attacks. Unfortunately, these techniques are not as robust as they seem and are broken by adaptive attacks [2, 13, 58]. Overall, to the best of our knowledge, there is little [40] to no evidence that data augmentations or generative models can be used to improve robustness to ℓ_p norm-bounded 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 [51]. We differentiate ourselves from earlier works by leveraging additional generated samples for training rather than modifying the defense procedure, and by establishing sufficient conditions under which such samples can improve robustness.

3 Adversarial training using generated data

The rest of this manuscript is organized as follows. In this section, we provide an overview of adversarial training, demonstrate using a motivational example that low-quality generated data can be leveraged to improve robustness to adversarial examples, and describe our method. In Sec. 4, we detail sufficient conditions that explain why generated samples can improve robustness and explore the limitations of our approach. In Sec. 5, we analyze three complementary and recent generative models in the context of our method. Finally, we provide experimental results in Sec. 6.

3.1 Adversarial training

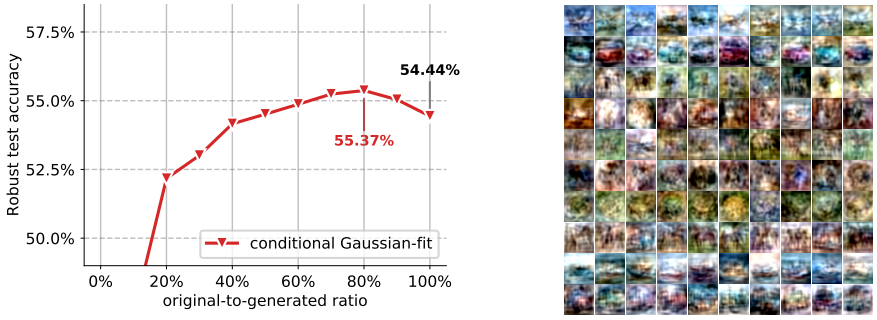
For classification tasks, Madry et al. [41] propose to find model parameters $\boldsymbol{\theta}$ that minimize the adversarial risk:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} \mathbb{1}_{f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta})\neq y} \right] \tag{1}$$

where $\mathcal{D}$ is a data distribution over pairs of examples $\boldsymbol{x}$ and corresponding labels y, $f(\cdot;\boldsymbol{\theta})$ is a model parametrized by $\boldsymbol{\theta}$, $\mathbb{1}$ is the indicator function and corresponds to the $0-1$ loss, and $\mathbb{S}$ defines the set of allowed perturbations. For ℓ_p norm-bounded perturbations of size ϵ, the perturbation set is defined as $\mathbb{S}_p = \{\boldsymbol{\delta} \mid \|\boldsymbol{\delta}\|_p \leq \epsilon\}$. Hence, for ℓ_∞ norm-bounded perturbations $\mathbb{S} = \mathbb{S}_\infty$ and for ℓ_2 norm-bounded perturbations $\mathbb{S} = \mathbb{S}_2$. In the rest of this manuscript, we use ϵ_p to denote ℓ_p norm-bounded perturbations of size ϵ (e.g., $\epsilon_\infty = 8/255$).

In practice, given a training set $\mathcal{D}_{\text{train}}$, the adversarial training procedure replaces the $0-1$ loss with the cross-entropy loss l_{ce} and is formulated as

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{train}}} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} l_{\text{ce}}(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}), y) \right]. \tag{2}$$



(a) (b)

Figure 3: Low-quality random inputs can improve robustness. Panel (a) shows the robust test accuracy (against AA+MT [25]) of a WRN-28-10 against $\epsilon_\infty = 8/255$ on CIFAR-10 trained with additional data randomly sampled from a class-conditional Gaussian fit of the training data. We compare how the ratio between original and generated images affects robustness (0% means generated samples only, while 100% means original CIFAR-10 train set only). Panel (b) shows some of the class-conditional Gaussian samples that are used during training.

3.2 Complementarity and diversity should improve robust generalization

Data augmentations can reduce the generalization error of standard (non-robust) training [15, 16, 20, 71]. However, to the contrary of standard training, augmentations beyond random flips, rotations and crops [27] – such as *Cutout* [20], *mixup* [71], *AutoAugment* [15] or *RandAugment* [16] – have been unsuccessful in the context of adversarial training [25, 52, 66]. The gap in robust accuracy between models trained with and without additional data (typically extracted from 80M-TI [57]) 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 [54].

We hypothesize that, to improve robust generalization, it is critical to use additional training samples (augmented or generated) that are diverse and that complement the original training set (in the sense that these new samples should ideally come from the same underlying distribution as the training set but should not duplicate the training set). To test this hypothesis, we propose to use samples generated from a simple class-conditional Gaussian fit of the training data. By construction, such samples (shown in Fig. 3(b)) 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 (no further filtering is applied). In Fig. 3(a), we show the performance of various robust models trained 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 an absolute improvement of +0.93%, which is an improvement comparable in size to the ones provided by model weight averaging or TRADES [25]. As we show in Sec. 4, the drop in robust accuracy for mixing ratios below 50% is expected in the capacity-limited regime. This experiment directly motivates our method.

3.3 Method

Given access to a pre-trained non-robust classifier f_{NR} and an unconditional generative model approximating the true data distribution $\mathcal{D}$ by a distribution $\hat{\mathcal{D}}$, we would like to train a robust classifier $f(\cdot;\boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$. We propose the following optimization problem:

$$\arg\min_{\boldsymbol{\theta}} \alpha \cdot \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{train}}} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} l_{\text{ce}}(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}), y) \right] + (1-\alpha) \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\hat{\mathcal{D}}} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} l_{\text{ce}}(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}), f_{\text{NR}}(\boldsymbol{x})) \right] \tag{3}$$

where α corresponds to the mixing ratio that blends examples from the training set with those that are generated. When α is set to one, our method reverts back to the original adversarial training

formulation in Eq. 2. When α is set to zero, our method only uses generated samples with their corresponding pseudo-labels. In practice, for efficiency, rather than generating samples on-the-fly, we pre-generate samples offline. Hence, both the original training set $\mathcal{D}_{\text{train}}$ and generated set $\hat{\mathcal{D}}$ contain a finite number of samples. We have the advantage, however, to be able to generate significantly more samples than present in the original training set. In App. B, we evaluate how varying the number of generated samples impacts adversarial robustness.

Overall, the complete method is described in Fig. 2 and is composed of three steps: *(i)* it starts by training the non-robust classifier and generative model on the original training set (for CIFAR-10, that corresponds to 50K images only); *(ii)* then, the generated dataset is constructed by drawing samples from the generative model and pseudo-labeling them using the non-robust classifier; *(iii)* finally, the robust classifier is trained using both the original training set and the generated dataset using Eq. 3.

4 Randomness might be enough

In this section, we formalize our notation (Sec. 4.1), and then provide three sufficient conditions that explain why generated data can improve robustness (Sec. 4.2). In summary, *(i)* the pre-trained, non-robust classifier f_{NR} used for pseudo-labeling must be accurate, *(ii)* the likelihood of sampling examples that are adversarial to this non-robust classifier must be low, and *(iii)* the generative model must be able to sample images from the true data distribution with non-zero probability.

4.1 Setup

Given a ground-truth function $f^\star$, we would like to find optimal parameters $\boldsymbol{\theta}^\star$ for $f(\cdot;\boldsymbol{\theta}^\star)$ that minimize the adversarial risk,

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} \mathbb{1}_{f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta})\neq f^\star(\boldsymbol{x})} \right], \tag{4}$$

without access to the true data distribution $\mathcal{D}$ or the ground-truth classifier $f^\star$. As such, we replace the distribution $\mathcal{D}$ with an approximated distribution $\hat{\mathcal{D}}$ (from a generative model) and use a pre-trained non-robust classifier f_{NR} instead of $f^\star$ (see Sec. 3.3). This results in sub-optimal parameters

$$\hat{\boldsymbol{\theta}}^\star = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim\hat{\mathcal{D}}} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} \mathbb{1}_{f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta})\neq f_{\text{NR}}(\boldsymbol{x})} \right]. \tag{5}$$

We introduce the unknown probability measure μ corresponding to the true data distribution $\mathcal{D}$ and defined over the set of inputs $\mathcal{A} \subseteq \mathbb{R}^n$ (where n is the input dimensionality), as well as the known probability measure $\hat{\mu}$ corresponding to the approximated distribution $\hat{\mathcal{D}}$. The set of relevant inputs $\mathcal{X} \subseteq \mathcal{A}$ (i.e., the set of *realistic* images for which we would like to enforce robustness) is the support of μ such that $\mu(\mathcal{X}) = 1$ and $\forall \mathcal{W} \subseteq \mathcal{X}, \mu(\mathcal{W}) > 0$. We assume that each input $\boldsymbol{x} \in \mathcal{X}$ can be assigned a label $y = f^\star(\boldsymbol{x})$ where $f^\star : \mathcal{X} \mapsto \mathcal{Y}$ is the ground-truth classifier (only valid for *realistic* images) and $\mathcal{Y} \in 2^{\mathbb{Z}}$ is the set of labels. Finally, given a perturbation set $\mathbb{S}$, we restrict labels such that there exists no *realistic* image within the perturbation set of another that has a different label; i.e., given an input $\boldsymbol{x} \in \mathcal{X}$, for all $\boldsymbol{\delta} \in \{\boldsymbol{\delta}' \in \mathbb{S} | \boldsymbol{x} + \boldsymbol{\delta}' \in \mathcal{X}\}$ we have $f^\star(\boldsymbol{x}) = f^\star(\boldsymbol{x} + \boldsymbol{\delta})$.

4.2 Limitations and sufficient conditions

To understand the limitations of our approach, it is useful to think about idealized sufficient conditions that would allow the sub-optimal parameters $\hat{\boldsymbol{\theta}}^\star$ to approach the performance of the optimal parameters $\boldsymbol{\theta}^\star$. First, we concentrate on the capacity-limited regime and later extrapolate to the infinite-capacity, infinite-compute regime to gain more insights. The first sufficient condition concerns the pre-trained non-robust classifier and holds for both regimes.

Condition 1 (accurate non-robust classifier)**.** *The pre-trained non-robust classifier $f_{NR} : \mathcal{A} \mapsto \mathcal{Y}$ must be accurate on all* realistic *inputs $\boldsymbol{x} \in \mathcal{X}$: $\forall \boldsymbol{x} \in \mathcal{X}, f_{NR}(x) = f^\star(x)$.*

Indeed, if we had access to the true distribution $\mathcal{D}$, Eq. 4 and Eq. 5 could be made equal by setting $f^\star(x) = f_{\text{NR}}(x)$. [1] In the capacity-limited regime, when Cond. 1 is satisfied, the problem reduces to

[1] In Sec. 6, we show that sub-optimal parameters $\hat{\boldsymbol{\theta}}^\star$ that improve upon those obtained by Eq. 2 can be obtained even when the non-robust classifier f_{NR} is not perfect. In our experiments, we use a classifier that achieves 95.68% on the CIFAR-10 test set.

a robust generalization problem. This problem is widely studied [3, 18, 61] and one can show that the adversarial risk is bounded by the Wasserstein distance between the training distribution $\hat{\mathcal{D}}$ and true data distribution $\mathcal{D}$ (under mild assumptions) [38]. In other words, as $\hat{\mathcal{D}}$ approaches $\mathcal{D}$, we expect the robust accuracy of $f(\cdot;\hat{\boldsymbol{\theta}}^\star)$ to approach the one of $f(\cdot;\boldsymbol{\theta}^\star)$. This intuitively leads to the second sufficient condition.

Condition 2 (accurate approximated distribution)**.** *The approximated data distribution $\hat{\mathcal{D}}$ and true data distribution $\mathcal{D}$ must be equivalent: $\mu(\mathcal{W}) = \hat{\mu}(\mathcal{W})$ for all measurable subset $\mathcal{W} \subseteq \mathcal{X}$.*

Together, Cond. 1 and 2 provide sufficient conditions for the capacity-limited regime. Cond. 2 (and associated bounds from [38]) generally indicates that the robust accuracy of the classifier $f(\cdot;\hat{\boldsymbol{\theta}}^\star)$ should increase as the quality of the generative model that provides the approximated distribution $\hat{\mathcal{D}}$ improves. However, these two conditions do not provide a satisfying answer when it comes to understanding why seemingly random data can help improve robustness (as demonstrated in Sec. 3.2). To help our understanding, it is worth analyzing the consequence of increasing the capacity of f. In particular, in the infinite-capacity regime, Cond. 2 can be relaxed and replaced by two conditions.

Condition 3 (unlikely adversarial examples)**.** *It is not possible to sample a point $\boldsymbol{x} \sim \hat{\mathcal{D}}$ outside the* realistic *set $\mathcal{X}$ such that it is adversarial to f_{NR}: $\hat{\mu}(W) = 0$ on the measurable subset $\mathcal{W} = \{x+\delta \mid x \in X, \delta \in S, f_{NR}(x+\delta) \neq f_{NR}(x)\}$.*

Condition 4 (sufficient coverage)**.** *There must be a non-zero probability of sampling any point from the* realistic *set: $\hat{\mu}(\mathcal{W}) > 0$ for all measurable subsets $\mathcal{W} \subseteq \mathcal{X}$.*

Cond. 3 enforces that labels are non-conflicting within the perturbation set of a *realistic* input,[2] while Cond. 4 guarantees that *realistic* inputs appear with enough frequency during training. Together, Cond. 1, 3 and 4 do not only provide sufficient conditions for the infinite-capacity regime, [3] but also explain why samples generated by a simple class-conditional Gaussian-fit can be used to improve robustness. Indeed, they 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, when compute and capacity are limited, it is critical that the optimization in Eq. 5 focuses on *realistic* inputs and that the distribution $\hat{\mathcal{D}}$ be as close as possible to the true distribution $\mathcal{D}$. In practice, this translates to the fact that better generative models (such as DDPM) can be used to achieve better robustness.

Impact of the mixing ratio α. Finally, we briefly address the impact of the mixing ratio α used in Eq. 3. Ignoring the change of loss, Eq. 3 can be formulated as Eq. 5 by using the non-robust classifier f'_{NR} instead of f_{NR} and using the merged distribution $\hat{\mathcal{D}}'$ instead of $\hat{\mathcal{D}}$, with

$$f'_{\text{NR}}(x) = \begin{cases} f^\star(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \mathcal{D}_{\text{train}} \\ f_{\text{NR}}(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{6}$$

and with $\hat{\mathcal{D}}'$ such that sampling $\boldsymbol{x} \sim \hat{\mathcal{D}}'$ is equivalent to $\boldsymbol{x} = \mathbb{1}_{r \leq \alpha}\boldsymbol{x}' + \mathbb{1}_{r > \alpha}\boldsymbol{x}''$ with $r \sim \mathcal{U}_{[0,1]}, \boldsymbol{x}' \sim \mathcal{U}_{\mathcal{D}_{\text{train}}}$ and $\boldsymbol{x}'' \sim \hat{\mathcal{D}}$ (where $\mathcal{U}_{\mathbb{A}}$ corresponds to the uniform distribution over set $\mathbb{A}$). This transformation artificially improves the accuracy of the non-robust classifier and reduces the gap between $\mathcal{D}$ and $\hat{\mathcal{D}}'$, thus resulting in better coverage. Note, however, that while increasing α improves the realism of training samples, it comes at the cost of a reduction in complementarity with the training set.

5 Generative models

The derivations from Sec. 4 and the experiment performed in Sec. 3.2 strongly suggest that generative models, which are capable of creating novel images [46], are viable augmentation candidates for adversarial training.

[2] Unless the generative model is trained to produce adversarial examples, random sampling is unlikely to produce images that are adversarial [7]. In fact, even strong black-box adversarial attacks require thousands of model queries to find adversarial examples.

[3] In this regime, the solution $f(\cdot;\boldsymbol{\theta}^\star)$ to Eq. 4 achieves perfect robust classification of all inputs $\boldsymbol{x} \in \mathcal{X}$ (since there are no conflicting labels, see Sec. 4.1). Similarly, when Cond. 1 and 3 are satisfied, the solution to Eq. 5 achieves perfect robust classification of all inputs $\boldsymbol{x} \in \hat{\mathcal{X}}$ where $\hat{\mathcal{X}} = \text{supp}(\hat{\mu})$ is the support of $\hat{\mu}$. As Cond. 4 enforces that $\mathcal{X} \subseteq \hat{\mathcal{X}}$, we have that $f(\cdot;\hat{\boldsymbol{\theta}}^\star)$ achieves perfect robust classification over $\mathcal{X}$.

Table 1: Complementarity and coverage of augmented and generated samples. We sample 10K images from the train set and various different generative models. For each sample in each set, we find its closest neighbor in Inception feature space (obtained after the pooling layer). To estimate complementarity, we report the proportion of samples with a nearest neighbor in either the train set, test set or the sampled set itself. To estimate coverage, we report the proportion of unique neighbors in the train and test set. We also include the IS and FID computed from 50K samples from each set.

	Complementarity			Coverage		Inception Metrics	
Setup	Train	Test	Self	Train	Test	IS ↑	FID ↓
Extracted from 80M-Ti [10]	30.86%	31.03%	38.11%	50.64%	50.41%	11.78 ± 0.12	2.80
mixup [71]	90.34%	3.91%	5.75%	90.43%	45.61%	9.33 ± 0.22	7.71
Class-conditional Gaussian-fit	0.13%	0.22%	99.65%	12.36%	12.24%	3.64 ± 0.03	117.62
VDVAE [11]	11.97%	12.14%	75.89%	34.20%	33.76%	6.88 ± 0.05	26.44
BigGAN [6]	14.97%	14.81%	70.22%	38.86%	39.06%	9.73 ± 0.07	13.78
DDPM [31]	29.29%	29.17%	41.54%	49.07%	49.10%	9.50 ± 0.14	3.15

Generative models considered in this work. 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 [6]: 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 [11]: a hierarchical Variational AutoEncoder (VAE) which outperforms alternative VAE baselines; and *(iii)* DDPM [31]: a diffusion probabilistic model based on Langevin dynamics that reaches state-of-the-art FID on CIFAR-10. [4] 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 App. D). Samples are shown in App. D.

Analysis of complementary and coverage. In Table 1, in addition to reporting IS and FID metrics, we evaluate how close Cond. 2 and Cond. 4 are to be satisfied in practice. To do so, we sample 10K images from each generative model. We also sample 10K images from the CIFAR-10 training set, and apply *mixup* to them as a point of comparison.[5] We observe that *mixup* achieves a similar IS to the BigGAN and DDPM models. In the left-most set of three columns, for each augmented or generated sample, we report whether its closest neighbor in the Inception[6] feature space belongs to the train set, test set or the generated set itself (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 each set. We observe that *mixup* tends to produce samples that are too close to the original train set and that lack complementarity, potentially explaining its 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. The middle two columns measure the ratio of unique neighbors that are matched in the train and test set. This provides a rough approximation of coverage. We observe a similar trend where samples from the DDPM seem to provide a better coverage of the true data distribution. Note that, while these numbers rely on an inaccurate distance measure (i.e., Euclidean distance in Inception feature space) and should be taken with a grain of salt, they correlate well with the results obtained from our experiments.

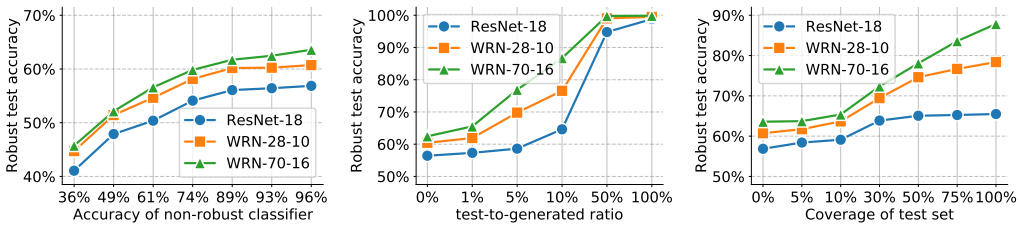
6 Experiments

The full experimental setup is explained in App. A. Specifically, we use Residual Networks (ResNets) and Wide ResNets (WRNS) [27, 69] with Swish/SiLU [29] activation functions. We use stochastic weight averaging [33] with a decay rate of $\tau = 0.995$. For adversarial training, we use TRADES [72]

[4] For VDVAE and DDPM, we use checkpoints available online. For BigGAN, we trained our own model.

[5] According to prior work, *mixup* is unable to improve robust accuracy beyond the one obtained with standard random cropping/flipping when using early stopping [52].

[6] Using the LPIPS [74] feature space (see Table 4 in the appendix) provides similar conclusions.



(a) Condition 1 (b) Condition 2 (c) Condition 4

Figure 4: Impact of violations of the sufficient conditions detailed in Sec. 4. We report the robust test accuracy (against AA+MT [25]) when training different model architectures against $\epsilon_\infty = 8/255$ on CIFAR-10 when using additional data produced by a DDPM. In panel (a), non-robust classifiers with different clean acccuracies are used for pseudo-labeling. In panel (b), we artificially inject samples from the test distribution at different frequencies (thus artificially closing the gap between generated and test distributions). In panel (c), we fix the test-to-generated ratio to 10% and restrict test samples to different subsets of the test set with different sizes (thus artificially changing the coverage).

with 10 Projected Gradient Descent (PGD) steps. We train for 800 CIFAR-10-equivalent epochs with a batch size of 1024 (i.e., 39K steps). We always evaluate our models against a combination of AUTOATTACK [13] and MULTITARGETED [24], which is denoted by AA+MT [25]. As a comparison point, we trained ten WRN-28-10 models on CIFAR-10. The resulting robust test accuracy on CIFAR-10 against $\epsilon_\infty = 8/255$ 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.

6.1 Sufficient conditions

The first set of experiments probes how violations of Cond. 1, 2 and 4 impact robustness against $\epsilon_\infty = 8/255$ on CIFAR-10 (we note that violations to Cond. 3 have an impact equivalent to those of Cond. 1). All experiments are summarized in Fig. 4 where we train three different model architectures with increasing capacity on 1M samples generated by the DDPM.

Non-robust classifier accuracy. In Fig. 4(a), we observe that robustness improves as the accuracy of the pre-trained, non-robust classifier f_{NR} increases. Notably, even with the 74.47%-accurate non-robust classifier, the WRN-20-10 and WRN-70-16 obtain robust accuracies of 58.15% and 59.83%, respectively, and already improve upon the state-of-the-art (57.14% at the time of writing) . Thus, validating that, in practice, it is not necessary to have access to a perfect non-robust classifier.

Quality of the generative models. To analyze how the quality of the generative model influences robustness, we artificially construct an idealized generative model that has access to test set samples. In particular, rather than always using DDPM samples, we also use test set samples chosen uniformly at random with at a given frequency, called the "test-to-generated" ratio. In effect, increasing the test-to-generated ratio towards 100% skews the generative model to produce more samples from the test set, thereby closing the gap between the approximated distribution $\hat{\mathcal{D}}$ and true distribution $\mathcal{D}$ (here being the distribution on which we measure the adversarial risk). As expected, Fig. 4(b) demonstrates that, given enough capacity, models can reduce the adversarial risk to zero.

Relationship between coverage and capacity. Coverage being difficult to measure, we idealize our generative model once again. In Fig. 4(c), we set the test-to-generated ratio to 10% (see previous paragraph) and only use a given fraction of test set samples. The coverage of the true data distribution $\mathcal{D}$ (approximated here by the test distribution) increases as this fraction increases. However, the approximated distribution $\hat{\mathcal{D}}$ remains different from the true data distribution even when the coverage reaches 100% (as the test-to-generated ratio is fixed to 10%). We observe that the robust accuracy of models with lower capacity plateaus, while the one of our largest model continues to improve. This observation confirms that, with enough coverage, model capacity can compensate for the lack of a perfect generative model.

6.2 State-of-the-art robust accuracy

Effect of mixing ratio (α). As done for the class-conditional Gaussian-fit in Sec. 3.3, we vary the ratio α of original-to-generated images in each batch for all three generated datasets. Fig. 5 explores a

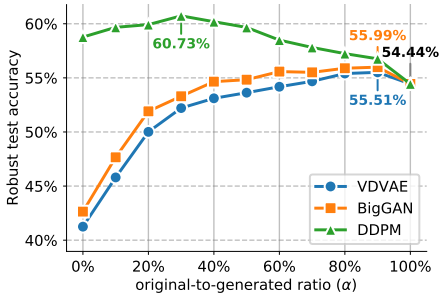


Figure 5: Robust test accuracy obtained by training a WRN-28-10 against $\epsilon_\infty = 8/255$ on CIFAR-10 when using additional data produced by different generative models. We compare how the ratio between original and generated images (i.e., α) affects robustness (0% means generated samples only, 100% means CIFAR-10 train set only).

MODEL	DATASET	NORM	CLEAN	ROBUST
Wu et al. [66] (WRN-34-10)	CIFAR-10	ℓ_∞	85.36%	56.17%
Gowal et al. [25] (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. [66] (WRN-34-10)	CIFAR-10	ℓ_2	88.51%	73.66%
Gowal et al. [25] (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. [17] (WRN-34-10)	CIFAR-100	ℓ_∞	60.64%	29.33%
Gowal et al. [25] (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 $\epsilon_\infty = 8/255$ and $\epsilon_2 = 128/255$. For CIFAR-100 and SVHN, we test against $\epsilon_\infty = 8/255$.

wide range of ratios while training a WRN-28-10 against $\epsilon_\infty = 8/255$ on CIFAR-10. A ratio of zero indicates that only generated images are used, while a ratio of 100% 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., [54]). 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. We also observe that the optimal mixing ratio is different for different generative models. Indeed, increasing α reduces the gap to the true data distribution at the cost of less complementarity with the original training set. In other words, when samples have low-diversity or low-quality, it is beneficial to use them less frequently.

CIFAR-10. Table 2 shows the performance of models trained with 1M samples generated by the DDPM on CIFAR-10 against $\epsilon_\infty = 8/255$ and $\epsilon_2 = 128/255$. 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, <u>despite not using any external data</u>, our best models beat all RobustBench [14][7] entries that used external data except one (see Table 5 in the appendix).

Generalization to other datasets (CIFAR-100 and SVHN). Finally, to evaluate the generality of our approach, we evaluate it on CIFAR-100 and SVHN [45]. We train two new DDPM on the train set of CIFAR-100 and SVHN and sample 1M images from each. The results are shown in Table 2. On CIFAR-100, our best model reaches a robust accuracy of 33.49% and improves noticeably upon the state-of-the-art by +3.46% (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, with the robust accuracy improving by by +4.07%.

7 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 current 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/[anonymized].

[7] https://robustbench.github.io/

References

[1] A. Athalye and I. Sutskever. Synthesizing robust adversarial examples. *Int. Conf. Mach. Learn.*, 2018.

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *Int. Conf. Mach. Learn.*, 2018.

[3] A. N. Bhagoji, D. Cullina, and P. Mittal. Lower bounds on adversarial robustness from optimal transport. *Adv. Neural Inform. Process. Syst.*, 2019.

[4] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*, 2013.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *NIPS Deep Learning Symposium*, 2016.

[6] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *Int. Conf. Learn. Represent.*, 2018.

[7] T. Brunner, F. Diehl, M. T. Le, and A. Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. *arXiv preprint arXiv:1812.09803*, 2019. URL https://arxiv.org/pdf/1812.09803.

[8] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

[9] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, 2017.

[10] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang. Unlabeled data improves adversarial robustness. In *Adv. Neural Inform. Process. Syst.*, 2019.

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

[12] P. Covington, J. Adams, and E. Sargin. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

[13] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *arXiv preprint arXiv:2003.01690*, 2020.

[14] F. Croce, M. Andriushchenko, V. Sehwag, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[15] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[16] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[17] J. Cui, S. Liu, L. Wang, and J. Jia. Learnable boundary guided adversarial training. *arXiv preprint arXiv:2011.11164*, 2020. URL https://arxiv.org/pdf/2011.11164.

[18] D. Cullina, A. N. Bhagoji, and P. Mittal. Pac-learning in the presence of adversaries. In *Adv. Neural Inform. Process. Syst.*, 2018.

[19] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin, G. v. d. Driessche, B. Lakshminarayanan, C. Meyer, F. Mackinder, S. Bouton, K. Ayoub, R. Chopra, D. King, A. Karthikesalingam, C. O. Hughes, R. Raine, J. Hughes, D. A. Sim, C. Egan, A. Tufail, H. Montgomery, D. Hassabis, G. Rees, T. Back, P. T. Khaw, M. Suleyman, J. Cornebise, P. A. Keane, and O. Ronneberger. Clinically applicable deep learning for diagnosis and referral in retinal disease. In *Nature Medicine*, 2018.

[20] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[21] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting Adversarial Attacks with Momentum. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *Int. Conf. Learn. Represent.*, 2015.

[23] S. Gowal, C. Qin, P.-S. Huang, T. Cemgil, K. Dvijotham, T. Mann, and P. Kohli. Achieving Robustness in the Wild via Adversarial Mixing with Disentangled Representations. *arXiv preprint arXiv:1912.03192*, 2019. URL https://arxiv.org/pdf/1912.03192.

[24] S. Gowal, J. Uesato, C. Qin, P.-S. Huang, T. Mann, and P. Kohli. An Alternative Surrogate Loss for PGD-based Adversarial Testing. *arXiv preprint arXiv:1910.09338*, 2019.

[25] S. Gowal, C. Qin, J. Uesato, T. Mann, and P. 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.

[26] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

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

[29] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[30] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. 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.

[31] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.*, 2021.

[32] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. GANSpace: Discovering Interpretable GAN Controls. *arXiv preprint arXiv:2004.02546*, 2020. URL https://arxiv.org/pdf/2004.02546.

[33] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging Weights Leads to Wider Optima and Better Generalization. *Uncertainty in Artificial Intelligence*, 2018.

[34] A. Jahanian, L. Chai, and P. Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/pdf?id=HylsTT4FvB.

[35] A. Krizhevsky, V. Nair, and G. Hinton. The CIFAR-10 dataset. 2014. URL http://www.cs.toronto.edu/kriz/cifar.html.

[36] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *ICLR workshop*, 2016.

[37] C. Laidlaw, S. Singla, and S. Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *Int. Conf. Learn. Represent.*, 2021. URL https://openreview.net/pdf?id=dFwBosAcJkN.

[38] J. Lee and M. Raginsky. Minimax statistical learning with wasserstein distances. *Adv. Neural Inform. Process. Syst.*, 2018.

[39] I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with warm restarts. In *Int. Conf. Learn. Represent.*, 2017.

[40] D. Madaan, J. Shin, and S. J. Hwang. Learning to generate noise for robustness against multiple perturbations. *arXiv preprint arXiv:2006.12135*, 2020. URL https://arxiv.org/pdf/2006.12135.

[41] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *Int. Conf. Learn. Represent.*, 2018.

[42] M. Mosbach, M. Andriushchenko, T. Trost, M. Hein, and D. Klakow. Logit Pairing Methods Can Fool Gradient-Based Attacks. *arXiv preprint arXiv:1810.12042*, 2018.

[43] A. Najafi, S.-i. Maeda, M. Koyama, and T. Miyato. Robustness to adversarial perturbations in learning from incomplete data. *Adv. Neural Inform. Process. Syst.*, 2019.

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

[45] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

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

[47] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020.

[48] T. Pang, X. Yang, Y. Dong, K. Xu, H. Su, and J. Zhu. Boosting Adversarial Training with Hypersphere Embedding. *Adv. Neural Inform. Process. Syst.*, 2020.

[49] A. Plumerault, H. L. Borgne, and C. Hudelot. Controlling generative models with continuous factors of variations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=H1laeJrKDB.

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

[51] S. Ravuri and O. Vinyals. Classification Accuracy Score for Conditional Generative Models. *arXiv preprint arXiv:1905.10887*, 2019. URL https://arxiv.org/pdf/1905.10887.pdf.

[52] L. Rice, E. Wong, and J. Z. Kolter. Overfitting in adversarially robust deep learning. *Int. Conf. Mach. Learn.*, 2020.

[53] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/pdf?id=BkJ3ibb0-.

[54] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially Robust Generalization Requires More Data. *Adv. Neural Inform. Process. Syst.*, 2018.

[55] L. Song, R. Shokri, and P. Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 241–257, 2019.

[56] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *Int. Conf. Learn. Represent.*, 2014.

[57] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008.

[58] F. Tramer, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020. URL https://arxiv.org/pdf/2002.08347.

[59] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble Adversarial Training: Attacks and Defenses. *arXiv preprint arXiv:1705.07204*, 2017. URL https://arxiv.org/pdf/1705.07204.

[60] F. Tramèr, J. Behrmann, N. Carlini, N. Papernot, and J.-H. Jacobsen. Fundamental Tradeoffs between Invariance and Sensitivity to Adversarial Perturbations. *arXiv preprint arXiv:2002.04599*, 2020. URL https://arxiv.org/pdf/2002.04599.

[61] Z. Tu, J. Zhang, and D. Tao. Theoretical analysis of adversarial learning: A minimax approach. In *Adv. Neural Inform. Process. Syst.* 2019.

[62] J. Uesato, B. O'Donoghue, A. v. d. Oord, and P. Kohli. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. *Int. Conf. Mach. Learn.*, 2018.

[63] J. Uesato, J.-B. Alayrac, P.-S. Huang, R. Stanforth, A. Fawzi, and P. Kohli. Are labels required for improving adversarial robustness? *Adv. Neural Inform. Process. Syst.*, 2019.

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

[65] B. Wu, J. Chen, D. Cai, X. He, and Q. Gu. Do wider neural networks really help adversarial robustness? *arXiv preprint arXiv:2010.01279*, 2021. URL https://arxiv.org/pdf/2010.01279.

[66] D. Wu, S.-t. Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. *Adv. Neural Inform. Process. Syst.*, 2020.

[67] C. Xie, Y. Wu, L. van der Maaten, A. Yuille, and K. He. Feature denoising for improving adversarial robustness. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[68] Y. Yang, G. Zhang, D. Katabi, and Z. Xu. Me-net: Towards effective adversarial robustness with matrix estimation. In *Int. Conf. Mach. Learn.*, 2019. URL http://proceedings.mlr.press/v97/yang19e/yang19e.pdf.

[69] S. Zagoruyko and N. Komodakis. Wide residual networks. *Brit. Mach. Vis. Conf.*, 2016.

[70] R. Zhai, T. Cai, D. He, C. Dan, K. He, J. Hopcroft, and L. Wang. Adversarially Robust Generalization Just Requires More Unlabeled Data. *arXiv preprint arXiv:1906.00555*, 2019.

[71] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *Int. Conf. Learn. Represent.*, 2018.

[72] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. *Int. Conf. Mach. Learn.*, 2019.

[73] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf?id=iAX0l6Cz8ub.

[74] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint arXiv:1801.03924*, 2018. URL https://arxiv.org/pdf/1801.03924.

[75] D. Zoran, M. Chrzanowski, P.-S. Huang, S. Gowal, A. Mott, and P. Kohl. Towards Robust Image Classification Using Sequential Attention Models. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We claim that adversarial robustness can be improved using generated data. Experiments demonstrate that this is possible and that the improvements are significant.
   (b) Did you describe the limitations of your work? [Yes] The sufficient conditions provided in Sec. 4 provide an idealized situation that we degrade in the experimental section. In particular, the method is intrinsically limited by the accuracy of the non-robust classifier f_{NR} as well as the ability to generate *realistic* inputs.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We highlight potential pitfalls in the introduction and related work. We summarize these and highlights them more broadly in App. F
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We provide two set of sufficient conditions. The first set, composed of Cond. 1 and Cond. 2, applies to the limited-capacity and limited-compute regime, while the second set, composed of Cond. 1, Cond. 3 and Cond. 2, applies to the infinite-capacity and infinite-compute regime. Other assumptions are details in Sec. 4.1.
   (b) Did you include complete proofs of all theoretical results? [Yes] Proof are included in the main text and rely on either demonstrating that Eq. 4 and Eq. 5 can be made equal when Cond. 1 and Cond. 2 are satisfied (limited-capacity) or showing that the combination of Cond. 1, Cond. 3 and Cond. 4 yields a perfectly robust classifier on the subset of inputs that includes the set of *realistic* inputs (infinite-capacity regime).
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The supplementary material includes evaluation code and links to relevant data and models. The training code is not shared at the moment as it is specific to our infrastructure. We do, however, provide all necessary hyper-parameters to reproduce our results.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All details are in App. A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Adversarial training is compute-intensive and the negative-impact of running more experiments outweighed the benefits. We use the same evaluation setup than proposed in [25]. That is, we always trained two models for each hyper-parameter setup and chose the best model according to a separate validation set of 1024 images (the average degradation in robustness between both models measured on a subset of 10 hyper-parameters setups is -0.12% in absolute robust accuracy). We also evaluated all models with one of the strongest set of adversarial attacks. Additionally, our baseline model was trained ten times resulting in a standard deviation of 0.39% and all reported improvements in robust accuracy are well beyond 2 standard deviations.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] All details are in App. A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We use CIFAR-10, CIFAR-100 and SVHN.
   (b) Did you mention the license of the assets? [No] Please refer to citations for details on licensing. All datasets are available for non-commercial use.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We generate additional data using generative models. These are available online and linked in the supplementary material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Experimental setup

Architecture. We use WRNs [27, 69] as our backbone network. This is consistent with prior work [25, 41, 52, 63, 72] which use diverse variants of this network family. Furthermore, we adopt the same architecture details as Gowal et al. [25] with Swish/SiLU [29] 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. Sven: Include training time on 4x4 TPUv3 and citation to it. Sven: Include JAX and citation to it.

Outer minimization. We use TRADES [72] optimized using SGD with Nesterov momentum [44, 50] 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 [39] without restarts where the initial learning rate is set to 0.1 and is decayed to 0 by the end of training (similar to [25]). We scale the learning rates using the linear scaling rule of Goyal et al. [26] (i.e., effective $\text{LR} = \max(\text{LR} \times \text{batch size}/256, \text{LR})$). We also use model weight averaging (WA) [33]. 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. [10] 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 [72]. This optimization procedure is done using the standard PGD formulation [36] with step-size $\epsilon/4$ and 10 steps.

Evaluation. Sven: No need for early stopping. We follow the evaluation protocol designed by Gowal et al. [25]. 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. [52] 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 [13] and MULTITARGETED [24], 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.

B Additional results

Scaling dataset size. Using a generative model allows us to sample many more images than available in the original training set. In Fig. 6, we set the mixing ratio α to zero (thus only using generated samples) and vary the number of training samples. We evaluate the robust accuracy of the resulting model on the test set and on a separate validation set composed of generated samples. We also compare models trained on BigGAN and DDPM samples. Irrespective of the generative model used, using more samples generally improves robustness. Samples from the DDPM model are more useful, as can be seen from the smaller generalization gap of 5.74 points when using 1M samples (gap between the dashed and solid blue lines). Models trained on BigGAN samples tend to overfit to these samples, which results in a large generalization gap of 51.61 points. These results also confirm that BigGAN samples are easier to robustly classify (possibly due to their low diversity).

CIFAR-100. For completeness, we also report the effect of mixing different proportions of generated and original samples in Fig. 7 against $\epsilon_\infty = 8/255$ using a WRN-28-10 on CIFAR-100. Similarly to Fig. 5, 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. 7 against $\epsilon_\infty = 8/255$ using a WRN-28-10 on SVHN. Similarly to Fig. 5 and Fig. 7, we observe that additional samples generated by DDPM are useful to improve robustness, with an absolute improvement of +4.07% in robust accuracy.

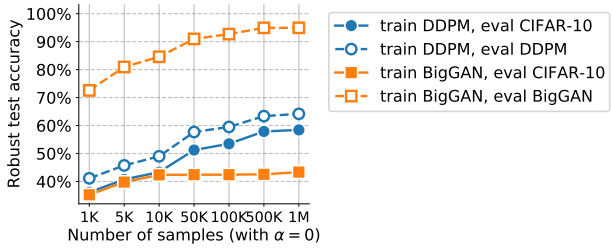


Figure 6: Robust test accuracy when training a WRN-28-10 using a variable number of samples from a DDPM or BigGAN. We compare the robust accuracy on the CIFAR-10 test set with the one obtained on a separate set of generated samples.

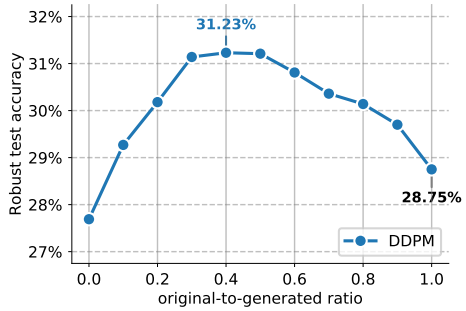


Figure 7: Robust test accuracy (against AA+MT) when training a WRN-28-10 against $\epsilon_\infty = 8/255$ on CIFAR-100 when using additional data produced by a DDPM. We compare how the ratio between original images and generated images in the training minibatches affects the test robust performance (0 means generated samples only, while 1 means original CIFAR-100 train set only).

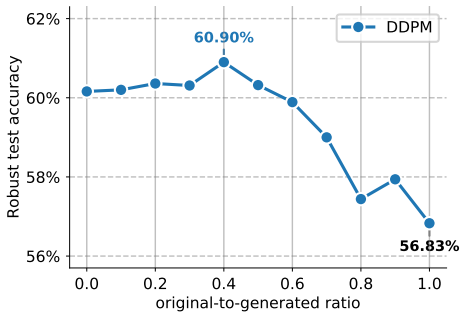


Figure 8: Robust test accuracy (against AA+MT) when training a WRN-28-10 against $\epsilon_\infty = 8/255$ on SVHN when using additional data produced by a DDPM. We compare how the ratio between original images and generated images in the training minibatches affects the test robust performance (0 means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

Clean accuracy. Sven: Add clean accuracy for the different mixing plots.

C Analysis of models

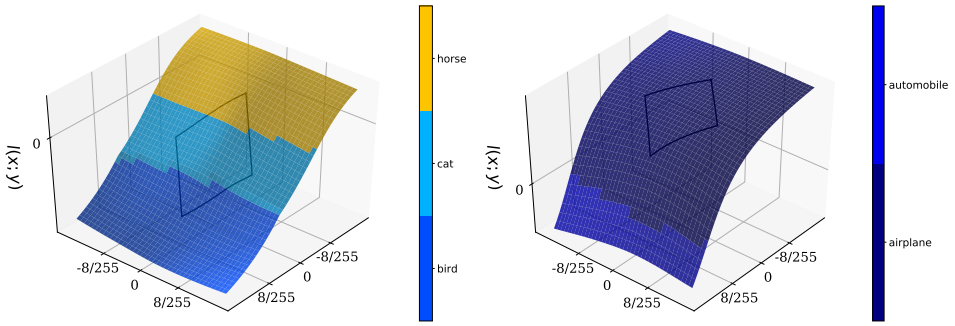
In this section, we perform additional diagnostics that give us confidence that our models are not doing any form of gradient obfuscation or masking [2, 62].

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.

Loss landscapes. We analyze the adversarial loss landscapes of our best model trained on CIFAR-10 against $\epsilon_\infty = 8/255$ (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 [9]: $z_y - \max_{i \neq y} z_i$ (i.e., a misclassification occurs when this value falls below zero). Fig. 9 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.

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.

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%



(a) Image of a horse

(b) Image of an airplane

Figure 9: 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 [6]: 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 [11]: a hierarchical VAE which outperforms alternative VAE baselines; and *(iii)* DDPM [31]: 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 6). 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 117.62 and 3.64, 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 3.15 and IS of 9.50.[8,9]

Datasets of generated samples. For the class-conditional multivariate Gaussian and DDPM samples, we use a pretrained WRN-28-10 to give pseudo-labels. This WRN-28-10 is trained non-robustly on the CIFAR-10 train set and achieves 95.68% accuracy.[10,11] We sample images from each model until we have 100K images for each class. For BigGAN and VDVAE samples, we proceed with an additional filtering simiarly to the one proposed by Carmon et al. [10]. We sample from each

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

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

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

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

Table 4: Complementarity and coverage of augmented and generated samples. **Dan:** maybe adding a descriptive title would help say why this table is important? (i.e., something about diversity and realism); the table actually makes at least two point now, maybe two subtables would be clearer? also, are the data augmentation rows detracting from the point about comparing additional external data with data coming from generative models? Essentially I feel this table "makes" the paper and I'm not sure it's clear enough now. 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 [74] 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. **Sven:** Update table with LPIPS and copy the one from main text.

	Neighbor Distribution				Inception Metrics	
Setup	Train	Test	Self	Entropy ↑	Is ↑	Fid ↓
Extracted from 80M-Ti [10]	30.12%	29.20%	40.68%	1.09	11.78 ± 0.12	2.80
mixup [71]	95.93%	0.32%	3.75%	0.18	9.33 ± 0.22	7.71
Cutout [20]	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 [11]	6.71 %	5.76%	87.53%	0.46	6.88 ± 0.05	26.44
BigGAN [6]	11.53%	10.51%	77.96%	0.68	**9.73**± 0.07	13.78
DDPM [31]	21.79%	20.16%	58.05%	9.50 ± 0.14	3.15	

generative model 5M images and score each image using the same pretrained WRN-28-10. For each class, we select the top-100K scoring images and build a dataset of 1M image-label pairs.[12]

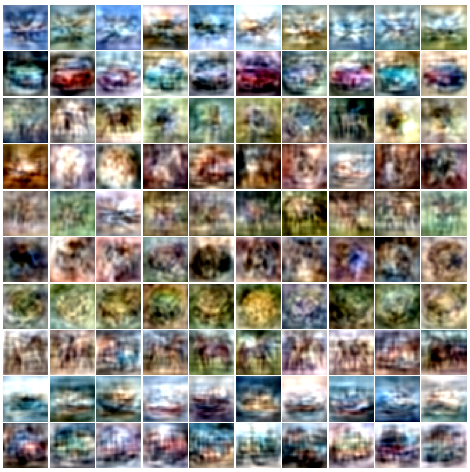
This additional generated data is used to train adversarially robust models by mixing for each batch a given proportion of original and generated examples. Fig. 10 shows a random subset of this additional data for each generative model. We also report the FID and IS metrics of the resulting sets in Table 1. They might differ from the metrics obtained by each generative model when we filter images to only pick the highest scoring ones.

Diversity and complementarity. **Sven:** Rewrite and mention LPIPS table (as opposed to Inception features).

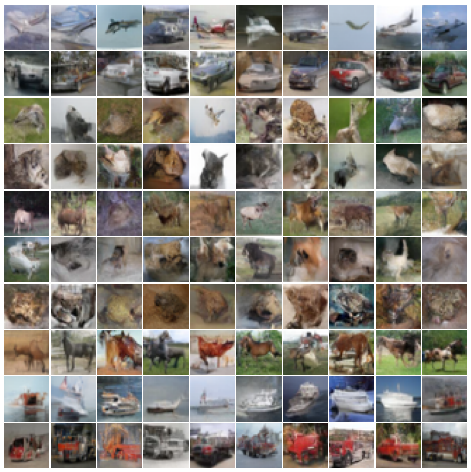
While the FID metric does capture how two distributions of samples match, it does not necessarily provide enough information in itself to assess the overlap between the distribution of generated samples and the train or test distributions (this is especially true for samples obtained through common data augmentations). As such, we also decide to compute the proportion of nearest neighbors in perceptual space. Given equal Inception metrics, a better 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 [74]. 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.

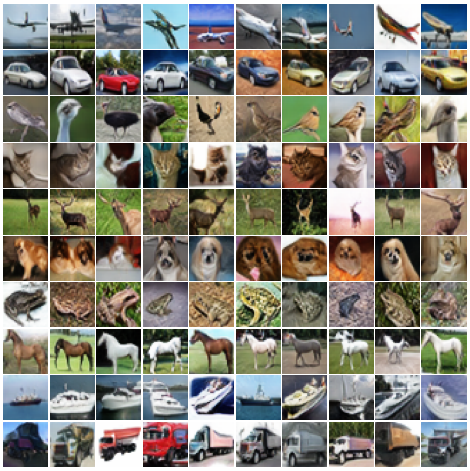
[12] All generated datasets are available online at https://github.com/[anonymized].



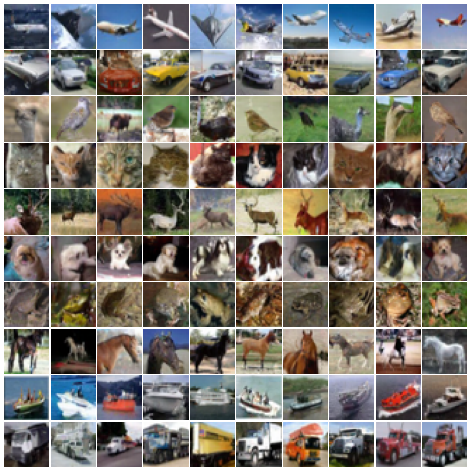
(a) Conditional Gaussian



(b) VDVAE



(c) BigGAN



(d) DDPM

Figure 10: 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.

E RobustBench

For reference, at the time of writing, the top-5 RobustBench (https://robustbench.github.io/ [14]) leaderboard entries without and with additional data are listed in Table 5. Entries from non-peer reviewed venues were only included if older than 2 months from writing.

F Societal impact

Sven: To do

Table 5: State of RobustBench leaderboard at the time of writing. We report the clean (without adversarial attacks) accuracy and robust accuracy on CIFAR-10 against $\epsilon_\infty = 8/255$.

AUTHOR	MODEL	CLEAN	ROBUST
WITHOUT EXTERNAL DATA			
Gowal et al. [25]	WRN-70-16	85.29%	57.14%
Gowal et al. [25]	WRN-34-20	85.64%	56.82%
Wu et al. [66]	WRN-34-10	85.36%	56.17%
Pang et al. [47]	WRN-34-20	86.43%	54.39%
Pang et al. [48]	WRN-34-20	85.14%	53.74%
WITH EXTERNAL DATA			
Gowal et al. [25]	WRN-70-16	91.10%	65.87%
Gowal et al. [25]	WRN-34-20	89.48%	62.76%
Wu et al. [65]	WRN-34-15	87.67%	60.65%
Wu et al. [66]	WRN-28-10	88.25%	60.04%
Zhang et al. [73]	WRN-28-10	89.36%	59.64%