ADAPTIVE GENERATION OF UNRESTRICTED ADVERSARIAL INPUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks are vulnerable to adversarially-constructed perturbations of their inputs. Most research so far has considered perturbations of a fixed magnitude under some l_p norm. Although studying these attacks is valuable, there has been increasing interest in the construction of—and robustness to—*unrestricted* attacks, which are not constrained to a small and rather artificial subset of all possible adversarial inputs. We introduce a novel algorithm for generating such unrestricted adversarial inputs which, unlike prior work, is *adaptive*: it is able to tune its attacks to the classifier being targeted. It also offers a 400–2,000× speedup over the existing state of the art. We demonstrate our approach by generating unrestricted adversarial inputs that fool classifiers robust to perturbation-based attacks. We also show that, by virtue of being adaptive and unrestricted, our attack is able to bypass adversarial training against it.

1 INTRODUCTION

Despite their dramatic successes in other respects, neural networks are well-known to not be adversarially robust. Szegedy et al. (2014) discovered that neural networks are vulnerable to what they termed *adversarial inputs*: by adding carefully-chosen perturbations to correctly-classified inputs, the accuracy of any neural network could be almost arbitrarily decreased. Since then, the machine learning community has rightly focused a great deal of research effort on this phenomenon. Many early efforts to train more robust models initially appeared promising, but have since been shown to be vulnerable to new algorithms for constructing adversarial perturbations (Xu et al., 2019). As a result, more attention has been given to methods that provide formal guarantees about performance in the presence of adversarial perturbations (Liu et al., 2019), with the state of the art now providing non-trivial guarantees for the MNIST test set (Wong & Kolter, 2018; Croce et al., 2018; Wang et al., 2018).

However, almost all of this work has focused exclusively on adversarial perturbations whose magnitude is constrained by an l_p norm. There is a growing acknowledgement that this threat model is somewhat contrived: such examples are not a realistic security concern and also occupy a vanishingly small fraction of the set of potential adversarial inputs. Therefore, there is a burgeoning interest in adversarial attacks that are *unrestricted*, in the sense that they do not necessarily derive from a perturbation of a natural input (Brown et al., 2018; Song et al., 2018b).

The main contribution of this paper is a novel and general method to generate unrestricted adversarial inputs. In short, the training procedure for generative adversarial networks (GANs) is modified so that the generator network is rewarded for producing data that are both realistic and deceive a fixed target network. Our approach has four advantages over prior work:

1. Our method is *adaptive* in that it adjusts itself to best attack the specific network being targeted. For instance, adversarial training is ineffective against our approach.
2. Our method is efficient (offering a 400–2000× speedup over prior work).
3. Our method can easily be applied to *any* existing GAN codebase and checkpoints, regardless of architecture, training procedure, or application domain.
4. Our method therefore demonstrably scales to ImageNet.

2 BACKGROUND: GENERATIVE ADVERSARIAL NETWORKS

Generative adversarial networks (GANs) (Goodfellow et al., 2014) are a class of generative machine learning models involving the simultaneous training of two neural networks: a generator g and a discriminator d. Specifically, given a dataset D of samples drawn from a probability distribution p_D, the generator g learns to transform random noise z drawn from a simple distribution p_z into an approximation of p_D. The discriminator network d learns to predict whether a given example x is drawn from the data distribution p_D or was generated by g. The generator and the discriminator are *adversarial* because they train simultaneously, with each being rewarded for out-performing the other.

GANs' training behaviours are notoriously temperamental, and many modifications to the original algorithm have been proposed (Goodfellow, 2017). The Wasserstein GAN variant (Arjovsky et al., 2017) aims to provide a more reliable gradient by designing the discriminator (renamed 'critic') to approximate the Wasserstein distance between the distribution generated by g_θ and the data distribution p_D. An additional 'gradient penalty' loss term L_{gp} can be added to implement the constraint that the function be 1-Lipschitz continuous. (Gulrajani et al., 2017) The loss functions for this Wasserstein GAN with gradient penalty (WGAN-GP) are: $L_g = \mathbb{E}_{z\sim p_z}[-d(g(z))]$ and $L_d = -L_g + \mathbb{E}_{x\sim p_D}[-d(x)] + \lambda L_{gp}$; where the gradient penalty $L_{gp} = \mathbb{E}_{\tilde{x}\sim p_I}[(\|\nabla_{\tilde{x}} d_\phi(\tilde{x})\|_2 - 1)^2]$, where p_I denotes the distribution sampling uniformly from the linear interpolations between generated samples and examples from p_D.

The original proposal for a conditional generative adversarial network (CGAN) learns to generate samples from a conditional distribution (Mirza & Osindero, 2014) by simply passing the intended label y for the generated image to both the generator and the discriminator. An extension of this approach is the auxiliary classifier generative adversarial network (ACGAN) (Odena et al., 2017), in which the discriminator is modified to also predict the label y for the input data. Both the generators are trained to maximise the log-likelihood of the correct label in addition to optimising their usual objective.

3 GENERATING UNRESTRICTED ADVERSARIAL INPUTS

Suppose we have a trained target classifier network $f: X \to \mathbb{R}^{|Y|}$ that attempts to approximate an oracle function $o\colon O \to Y$ (where $O \subseteq X$ is the oracle's domain) by outputting a confidence $f(x)_c \in \mathbb{R}$ for each class $c \in Y$. As Song et al. (2018b) do, we define an unrestricted adversarial example to be any input $x \in O$ such that the classifier's prediction is incorrect: $\text{argmax}_c f(x)_c \neq o(x)$. Unlike Song et al., we consider the domain of the oracle to be any input with a recognisable class, not just realistic inputs. This means we would still consider an unrealistic but recognisable image an unrestricted adversarial example. Nevertheless, we do carefully evaluate how realistic our results are in Section 4.1.

Unrestricted adversarial examples are a superset of conventional perturbation-based adversarial examples (which are restricted to lie within a fixed distance of some correctly-classified input from a test dataset). While providing a vastly larger space of candidates, a difficulty arises in determining that the classification is incorrect; we can no longer rely on the oracle-provided labels from the test dataset. We leverage generative models to solve this problem.

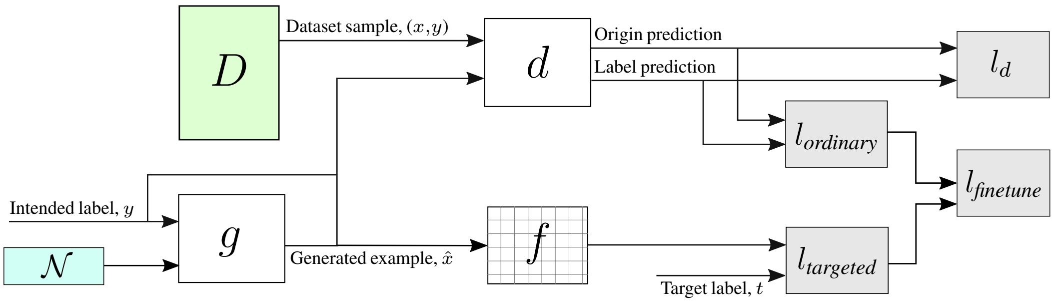


Figure 1: Diagram showing main data paths in the forward computation of loss functions. During adversarial finetuning, the generator and discriminator are trained by backpropagating the gradients from l_d and $l_{finetune}$ respectively. The target classifier, f, remains fixed.

3.1 OUR PROCEDURE

We train a class-conditional GAN to generate unrestricted adversarial inputs. This is achieved by simultaneously minimising an ordinary GAN loss and a new loss term. For an untargeted attack this term rewards the generator if the examples it generates are misclassified by the target network: $l_{untargeted} = f(\hat{x})_y - \max_{c \neq y} f(\hat{x})_c$. For a targeted attack, this term rewards the generator if the generated examples are assigned the desired target label, t, by the target classifier: $l_{targeted} = \max_{c \neq t} f(\hat{x})_c - f(\hat{x})_t$. Note that besides improving the quality of the generated data, our use of a conditional GAN and optimising for its ordinary loss function allows these new loss terms to be computed—otherwise, there is no way of knowing the label, y, for each generated image. This assumes that the true labels of the generated data match the intended labels passed as inputs to the generator, an assumption empirically validated in Section 4.1.

3.2 CHALLENGE: CONFLICTING GRADIENTS

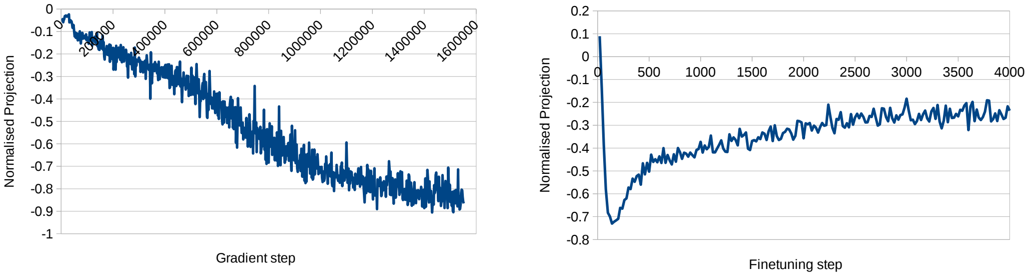
Naïvely optimising the sum of the two loss terms cripples training. There is no guarantee that the loss landscape will allow gradient-descent algorithms to find optimum where the images are both sufficiently realistic and adversarial, and unfortunately note that making an image adversarial seems likely to make the image *less* realistic, not more. This gives some intuition that the gradient from $l_{ordinary}$ may be pointing in a different direction to the gradient from $l_{(un)targeted}$.

A simple experiment suffices to verify this intuition. At each training step, the gradient vectors from both loss terms were normalised, then projected one onto the other. That is, the scalar quantity $\frac{\nabla l_{ordinary} \cdot \nabla l_{(un)targeted}}{||\nabla l_{ordinary}|| \, ||\nabla l_{(un)targeted}||}$ was computed. Figure 2a shows that this projection tends towards -1; for reference, if the gradient vectors were selected uniformly at random, the magnitude of this projection would rarely exceed 0.001. In other words, as training progresses, the gradients from these terms tend towards pointing in *opposite* directions. This makes joint optimisation using gradient descent a challenge.

3.3 STRATEGIES TO OVERCOME TRAINING CHALLENGES

Realistic pretraining It is widely accepted that real image data occupy a relatively low-dimensional and contiguous manifold (Goodfellow et al., 2016, p. 160). Conversely, we know that adversarial examples pervade the full input space: it appears that there is an adversarial example nearby nearly any point in the input space. Therefore, a generator that is pretrained using only $l_{ordinary}$ before *adversarially finetuning* by introducing our additional loss term is more successful than using both loss terms from a random initialisation. By beginning our search for unrestricted adversarial examples in regions of realistic examples, it is more likely that there are global optima of realistic adversarial examples nearby. Besides the generated images being subjectively better, Figure 2b shows that the gradients conflict to a much lesser extent. Note that *any* existing GAN architecture, pretrained checkpoint and training algorithm could be used here, allowing our method to leverage the significant advances being made in this area.

Amalgamation of loss terms The most naïve approach to jointly optimising an ordinary GAN loss term $l_{ordinary}$ with our additional loss term $l_{(un)targeted}$ is to simply minimise their sum. Part of the problem is that both terms continue to be minimised even if either one is 'good enough'; the generator



(a) Beginning from a randomly-initialised GAN.

(b) Adversarially finetuning a pretrained GAN.

Figure 2: Projecting normalised gradient vectors from $l_{ordinary}$ and $l_{(un)targeted}$ onto one another.

would work on making a misclassified example more strongly misclassified, which is not desirable. We use the following per-example loss function:

$$l_{finetune} = s(l_{ordinary}) \cdot s(l_{(un)targeted} - \kappa), \text{where } s(l) = \begin{cases} 1+\exp(l) & \text{if } l \leq 0, \\ 2+l & \text{otherwise.} \end{cases}$$

Here, κ is a hyperparameter similar to that in the Carlini & Wagner (2017) attack: it controls the confidence of the generated adversarial examples. If the difference between the desired logit and the next-greatest logit is less than κ, the generator is linearly rewarded for improving this gap (gaining confidence); beyond a difference of κ (once an example is 'good enough'), the reward exponentially decreases. We use $\kappa = 0$ for our experiments.

Stochastic loss selection The gradients from the two loss terms are in conflict, and so there is a danger that one of these may dominate the other. The generator may learn to generate unrealistic adversarial examples, which we found occurs in practice. The proportion of misclassified generated inputs rises quickly to almost 100%, but the generated images were noticeably unrealistic. This is a problem as their correct label may change. To address this, we introduce the 'attack rate' μ. During adversarial finetuning, the finetuning loss function is used only with probability μ; with probability $1-\mu$, the pretraining loss function ($l_{ordinary}$ only) is used. These gradient steps can be used to escape local optima of unrealistic adversarial examples. As desired, this new hyperparameter allows the success rate of the generated examples that are adversarial to be traded off with their visual quality.

4 EXPERIMENTAL EVALUATION

Our method aims to generate unrestricted adversarial inputs in a way that adapts to the targeted classifier. We therefore conducted experiments to check whether the generated examples were in fact unrestricted, adversarial, realistic, and adapting to the classifier. We then address some questions regarding the performance and generality of our approach.

The experimental evaluation is primarily on the MNIST dataset (LeCun et al., 1998), because this is the most challenging domain for the generation of realistic adversarial inputs. State-of-the-art classifiers perform very well, with around 0.2% test error (Kowsari et al., 2018; Wan et al., 2013). In particular, attempts to create robust classifiers have also been most successful on this dataset, perhaps due to its simplicity (Shafahi et al., 2018). We target five pretrained classifiers provably robust to adversarial perturbations: there is guaranteed to be no adversarial input within a distance ϵ of p% of test inputs under the l_∞ norm. All five are the current state-of-the-art in this domain, trained by Wong & Kolter (2018), and Wang et al. (2018). See Appendix C for details.

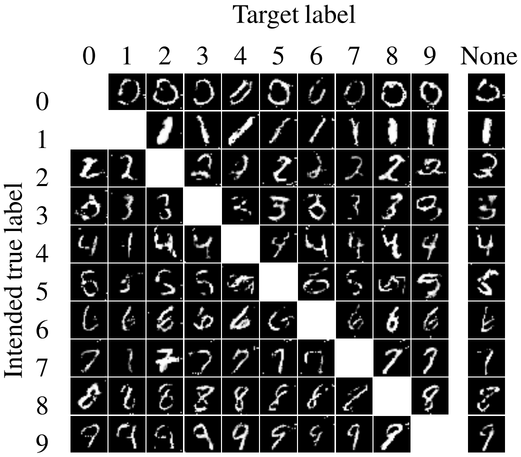


Figure 3: Randomly-selected images generated by a finetuned GAN to attack Wong and Kolter's locally-robust classifier.

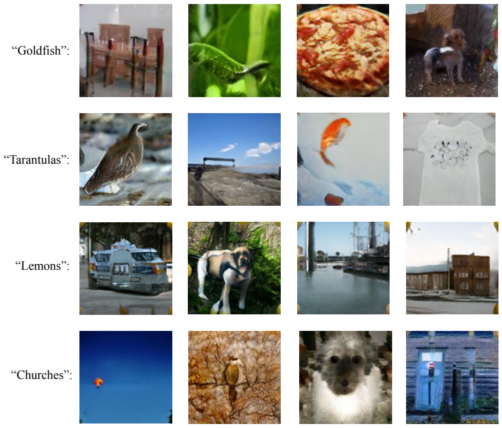
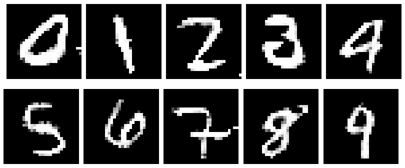


Figure 4: Selected ImageNet unrestricted adversarial examples.

Table 1: Comparison of typical perturbation magnitudes from the literature and ours.

Metric	Nearest neighbour seen	Typical perturbation magnitude
l_0	508	<40 (Ruan et al., 2018)
l_1	22.8	<5 (Lu et al., 2018)
l_2	3.28	~1.5 (Schott et al., 2018)
l_∞	0.838	~0.1 (Wong & Kolter, 2018)

Table 2: Ten selected unrestricted adversarial inputs used for Table 1.



In our experiments, we combine three well-established architectures: a Wasserstein GAN with gradient penalty (WGAN-GP) (Gulrajani et al., 2017), a conditional GAN (Mirza & Osindero, 2014) and an auxiliary classifier GAN (Odena et al., 2017). The generator is a convolutional neural network, conditioned on class label. The discriminator is a convolutional neural network with two separate, diverging final dense layers: one acts as a conditional WGAN-GP critic, the other as an auxiliary classifier. The auxiliary classifier helps the training converge, but is not necessary. Full details are given in Appendix E.

For each of the ten possible target labels—plus the untargeted case, which aims for any misclassification—a GAN was adversarially finetuned. After training converged, the generators were used to produce examples for all intended true labels, which were then filtered so that the computed label matched the target. Images were generated until 200 such filtered examples were generated for each intended true label/target label pairing or until 100 seconds had elapsed. Interestingly, this led to no adversarial examples with intended true label '0' and target classification '1', so this case is omitted. Figure 3 and Appendix B give examples of generated images for which the computed label matches the target classification.

4.1 EFFICACY OF ATTACKS

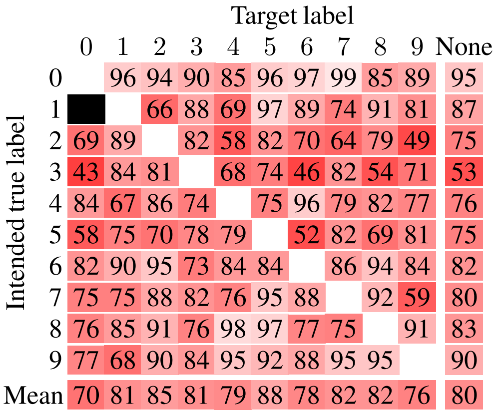
We claim that our method generates unrestricted adversarial examples, which are somewhat realistic. We empirically verify each claim in turn.

Since our method does not work by perturbing existing data, only a simple sanity check was required to verify that the generated images are not close to images in the training set, as could be caused by over-fitting. We selected ten generated inputs that are visually similar to the training set, and computed the shortest distances between the images and all images in the training set. The selected images are given in Figure 2. Table 1 shows that they are much further from any training example than would be the case with a perturbation-based attack.

Next, we evaluate the whether our method is successful in generating adversarial inputs. Our method generates such inputs if and only if their true label matches their intended true label given as input to the generator, else the generator could simply be ignoring this input and generating images which visually match the target class. To check that the finetuned generators are behaving as hoped, we used workers on Amazon's MTurk platform to label the generated images. For cost reasons, we only carried out the MTurk experiments targeting Wong and Kolter's provably-robust network (Wong & Kolter, 2018), not any of the MixTrain models. We used a sample size of 100 judges for each intended true label/target label pair for each experiment. Figure 5 shows the proportion of inputs for which not only does the label computed by the classifier match the target label, but the human-judged true label matches the intended true label specified to the generator. The mean number of correct labels for the untargeted attack is 80%. This high rate reassures us that our method really does generate successful unrestricted adversarial inputs.

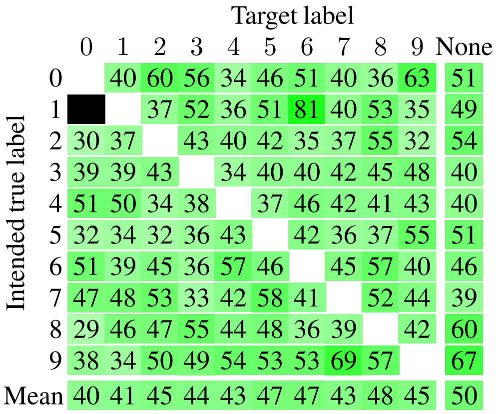
We now investigate if the generated examples are realistic. A set of inputs is *realistic* with respect to a dataset if a human cannot reliably identify to which set an example belongs. To check this, we again used MTurk workers. After familiarising themselves with examples from the training dataset, each worker had to pick which image was most likely to have been computer-generated out of ten images. Figure 6 shows the proportion of the time that generated images were not identified as such.

For comparison, we repeated these experiments but attacking a non-robust classifier network. The untargeted success rate was 90% (vs. 80% against the robust classifier), and 60% (vs. 50%) were not identified as being generated. Similar differences were seen in targeted attacks; see Appendix I.



Intended true label	Target label										
	0	1	2	3	4	5	6	7	8	9	None
0		96	94	90	85	96	97	99	85	89	95
1	■		66	88	69	97	89	74	91	81	87
2	69	89		82	58	82	70	64	79	49	75
3	43	84	81		68	74	46	82	54	71	53
4	84	67	86	74		75	96	79	82	77	76
5	58	75	70	78	79		52	82	69	81	75
6	82	90	95	73	84	84		86	94	84	82
7	75	75	88	82	76	95	88		92	59	80
8	76	85	91	76	98	97	77	75		91	83
9	77	68	90	84	95	92	88	95	95		90
Mean	70	81	85	81	79	88	78	82	82	76	80

Figure 5: The success rates of the adversarial attacks by finetuned GANs (the computed label matches the target label and the true label remains the same).



Intended true label	Target label										
	0	1	2	3	4	5	6	7	8	9	None
0		40	60	56	34	46	51	40	36	63	51
1	■		37	52	36	51	81	40	53	35	49
2	30	37		43	40	42	35	37	55	32	54
3	39	39	43		34	40	40	42	45	48	40
4	51	50	34	38		37	46	42	41	43	40
5	32	34	32	36	43		42	36	37	55	51
6	51	39	45	36	57	46		45	57	40	46
7	47	48	53	33	42	58	41		52	44	39
8	29	46	47	55	44	48	36	39		42	60
9	38	34	50	49	54	53	53	69	57		67
Mean	40	41	45	44	43	47	47	43	48	45	50

Figure 6: How often adversarial images are not identified as being generated. If the generated images were completely realistic, the expected result would be 90.

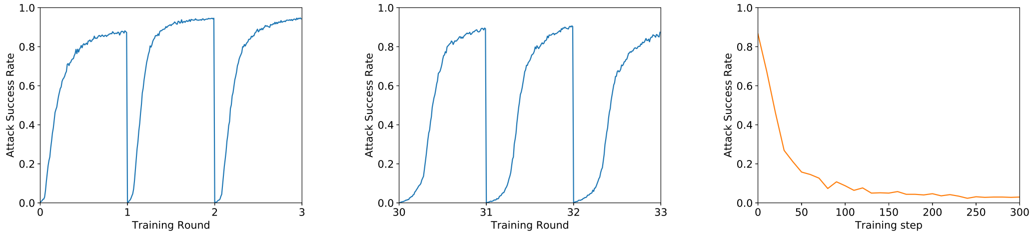
4.2 ADAPTIVITY TO ADVERSARIAL TRAINING DEFENCES

In the previous experiments we have tested our model against pretrained classifiers, which were provably robust to adversarial perturbations. We investigate whether standard adversarial training (Madry et al., 2018) against our attack in particular is effective. Starting with a pretrained GAN and classifier, we iterate 'training rounds' consisting of two phases. First, a GAN is adversarially finetuned (starting from the pretrained GAN each time) for a fixed period to attack the classifier. Second, 80,000 generated unrestricted adversarial examples are added to the existing training dataset, and the classifier *continues* training until almost 100% accuracy is achieved.

Figure 7a shows that, for the first few training rounds, adversarial finetuning is successful: the proportion of examples generated which fool the classifier increases to over 80%. Figure 7b shows the same story 30 rounds (and hence hundreds of thousands of classifier gradient steps) in. Although the classifier may be able to defend against the kinds of attacks learnt by the generator in previous training rounds, the generator's opportunity to adversarially finetune again allows it to generate adversarial examples of a kind not seen before by the classifier. Since the generator is unrestricted, it seems unlikely to 'run out' of these. For more details on these experiments, see Appendix D.

4.3 SCALING TO IMAGENET

While the MNIST classifiers are the most challenging to fool, MNIST is a relatively small and simple dataset. To demonstrate the scalability of our method, we apply it to the notoriously large and complex ImageNet-1K dataset. We also take advantage of the fact that our method works with *any* pretrained GAN by using the author's 'officially unofficial' published code and checkpoints for the current state-of-the-art, BigGAN (Brock et al., 2019). In the untargeted case, our method is able to finetune this BigGAN to fool the classifier >99% of the time within 40 gradient steps (compared to the 10^5 taken to train from scratch). Our main focus, though, is on the much more challenging targeted attack. We finetuned a BigGAN several times, selecting a variety of target classes. We found that typically, on the order of 100 gradient steps were required for >10% of generated examples to be classified (top-1) as the target class. Compared to MNIST, each ImageNet gradient step takes about 100x longer to compute, but the 100x decrease in the number of gradient steps required compensates for this, resulting in a similar compute time overall. Image quality as measured by Inception Score (Salimans et al., 2016) typically decreased from 70, slightly better than mid-2018 state-of-the-art of 52 (Zhang et al., 2019) to the mid-2017 state-of-the-art, WGAN-GP, of 12 (Shmelkov et al., 2018). We speculate that if the GAN were finetuned for significantly longer, the gradient from the discriminator would learn to regain some of this lost realism. Figure 4 shows selected samples of generated adversarial examples; Appendix A has a more extensive collection.



(a) Adversarial training against our attack, first few training rounds.

(b) Adversarial training against our attack, later training rounds.

(c) Adversarial training against Song et al. (2018b).

Figure 7: Plots showing attack efficacy in the presence of adversarial training.

4.4 ABLATION STUDY

To determine the contribution of our method, the MNIST experiments described were rerun but using a GAN not adversarially finetuned. Unsurprisingly, the desired misclassifications occurred vastly less frequently than when generated by a finetuned GAN. Furthermore, of those which were misclassified, the proportion for which the true label also matched the intended true label was also significantly lower without adversarial finetuning: 66% for untargeted attacks and 58% on average for targeted attacks, compared to 80% for both categories after finetuning. Full results are given in Appendix K. In short, we found that finetuning a GAN using our method roughly maintains how realistic its generated expected-adversarial images are, while significantly increasing their quantity and increasing the proportion which are true adversarial inputs by around 15 percentage points.

4.5 THREATS TO VALIDITY

The evaluation of the success of the attacks relies relies on data provided by the MTurk workers. We therefore employed measures to safeguard the quality of this data, described in Appendix H. We also believe that our method will generalise to any dataset and domain for which GANs can be trained successfully. However, this has only been demonstrated on two image classification tasks (albeit dissimilar in nature). Lastly, intuition suggests that our method will be able to adapt to find unrestricted adversarial examples for almost any defence method, since it is so free to generate inputs without the constraints that current defence methods rely upon. However, we have only demonstrated it explicitly for the most popular standard defence; future work may find a defence against our approach.

5 RELATED WORK

5.1 COMPARISON TO THE STATE OF THE ART

We compare our method to that of Song et al. (2018b), the current state of the art in generating unrestricted adversarial examples. Like ours, this method leverages a pretrained GAN. It differs, however, in how adversarial examples are then produced. Instead of adversarially finetuning the generator, it searches for an input to the generator that both deceives the target network and are confidently correctly classified by the discriminator's auxiliary classifier (an ACGAN (Odena et al., 2017) is required in this case). The GAN training is therefore blind to the target classifier.

Our model achieves similar success rates in generating unrestricted adversarial examples. In Section 4.1 we found that 80% of our generated inputs were 'adversarial' (the targeted neural network was incorrect). This is roughly comparable to Song et al.'s 88.8%. We also repeated the realism experiments from Section 4.1, with the difference that judges were asked to identify the one generated image out of a choice of two. In this case, Song et al. report that participants select the generated image as the more realistic 21.8% of the time; for our untargeted attack, this figure is 24%. The methods therefore produce similarly-realistic examples. Full results are given in Appendix J.

Beyond achieving comparable attack success rates, our approach has four significant advantages over prior work. Firstly: adaptivity. In Section 4.2 we have shown that our model is capable of iteratively adapting to an adversarially trained classifier. By contrast, Song et al.'s method performs poorly against adversarial training because the GAN is not trained with respect to a target classifier, remaining fixed after the attack begins. Therefore, if a classifier learns to be correct in the space their algorithm searches, it will no longer be able to generate images different enough to be adversarial. Figure 7c shows that standard adversarial training easily defends against Song et al.'s attack, while it fails against ours. Secondly: efficiency. Once trained, our method requires only a single forward pass to generate adversarial examples. Song et al. require 100–500 iterations, each with forward and backward passes through both the generator and classifier. Our method is therefore 400–2,000× more efficient. Lastly: scale and versatility. Section 4.3 shows that our model scales to ImageNet, a dataset with dimensionality 16× greater than the largest Song et al. demonstrate on. Our method has the further benefit that we can use any pretrained GAN, such as BigGAN (Brock et al., 2019). Song et al. depend on an auxiliary classifier for larger datasets, which BigGAN does not provide.

5.2 OTHER RELATED WORK

Wang et al. (2019) independently propose a method which is superficially similar to ours: they also train a GAN to directly generate adversarial examples. However, instead of using the ordinary GAN loss to ensure that the adversarial examples are sufficiently realistic, they instead use a new loss term. This term, $\|g_{pretrained}(z) - g(z)\|_p$, penalises the generator g given input z proportional to the deviation of its output from what it would have output immediately before adversarial finetuning. Our approach, to use the ordinary GAN loss for this purpose, allows for truly unrestricted adversarial examples, giving the training procedure much more scope to adapt to circumvent any specific defences (such as robustness to l_p perturbations). Wang et al.'s choice of loss term has the unfortunate effect of preventing the generator from generating either unrestricted adversarial examples or examples which are sure to fall within an l_p-norm ball of a realistic input. Our method has three further advantages over this work: we evaluate against state-of-the-art provably-robust networks rather than ad hoc classifiers; we conduct a user study to quantitatively verify the proportion of generated adversarial examples which maintain the correct label rather than merely assuming that this is 100%, which is unlikely; and we demonstrate that our approach works beyond MNIST scale (to ImageNet).

Sharif et al. (2019) train a network to generate patterned spectacles, which, when added to an image of a face, cause misclassification. They also adapt this approach to generate unrestricted adversarial examples for MNIST using an approach quite similar to ours. However, this only achieves a success rate of 8.34% against a classifier which was state-of-the-art in 2017, which is reduced to 0.83% after filtering to "'only the digits that where likely to be comprehensible by humans'. In contrast, we achieve around 80% accuracy against current state-of-the-art robust classifiers.

A wide range of work trains networks to generate adversarial *perturbations* (Hayes & Danezis, 2018; Baluja & Fischer, 2018; Xiao et al., 2018; Song et al., 2018a; Poursaeed et al., 2018). While these must also balance realism and adversarial success, the key difference is that we generate *unrestricted* adversarial examples, allowing attacks to succeed when constrained perturbations provably fail.

Hu et al. (2019) introduce a search for pairs of nearby unrestricted adversarial examples, but unfortunately cannot ensure that their true label is meaningful; if the search starting point is random, it is overwhelmingly likely not to be. If instead it is a known input, the examples are not unrestricted.

6 CONCLUSION

We have introduced an algorithm which trains a GAN to generate unrestricted adversarial inputs; we demonstrate that these, as expected, are successful against state-of-the-art classifiers robust to perturbation attacks. The key novelty in our attack procedure is that it entails the tuning of the weights of the generator to target a specific network. As a result, it can be considered *adaptive*: we have shown that, while prior work is quickly mitigated by standard adversarial training, our attack adapts to find a new way of fooling the classifier In addition, once the generator is adversarially finetuned, it becomes an endless supply of cheap adversarial examples: generation of an adversarial example requires a single forward pass rather than execution of any optimisation algorithm, resulting in a 400–2000× speedup over the state of the art. We have also demonstrated that any existing GAN codebase can easily be used by adapting BigGAN to generate unrestricted adversarial examples for ImageNet.

REFERENCES

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh (eds.), *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223. PMLR, 2017. URL `http://proceedings.mlr.press/v70/arjovsky17a.html`.

Shumeet Baluja and Ian Fischer. Learning to attack: Adversarial transformation networks. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *AAAI Conference on Artificial Intelligence*, pp. 2687–2695. AAAI Press, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16529`.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=B1xsqj09Fm`.

Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Francis Christiano, and Ian J Goodfellow. Unrestricted adversarial examples. *CoRR*, abs/1809.0, 2018. URL `http://arxiv.org/abs/1809.08352`.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP*, pp. 39–57. IEEE Computer Society, 2017. ISBN 978-1-5090-5533-3. doi: 10.1109/SP.2017.49. URL `https://doi.org/10.1109/SP.2017.49`.

Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of ReLU networks via maximization of linear regions. *CoRR*, abs/1810.07481, 2018. URL `http://arxiv.org/abs/1810.07481`.

Ian J Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.0, 2017. URL `http://arxiv.org/abs/1701.00160`.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D Lawrence, and Kilian Q Weinberger (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2672–2680, 2014. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets`.

Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016. ISBN 978-0-262-03561-3. URL `http://www.deeplearningbook.org/`.

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M Wallach, Rob Fergus, S V N Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5769–5779, 2017. URL `http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans`.

Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops*, pp. 43–49. IEEE Computer Society, 2018. doi: 10.1109/SPW.2018.00015. URL `https://doi.org/10.1109/SPW.2018.00015`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034. IEEE Computer Society, 2015. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.123. URL `https://doi.org/10.1109/ICCV.2015.123`.

Hanbin Hu, Mit Shah, Jianhua Z. Huang, and Peng Li. Global adversarial attacks for assessing deep learning robustness. *CoRR*, abs/1906.07920, 2019. URL `http://arxiv.org/abs/1906.07920`.

Kamran Kowsari, Mojtaba Heidarysafa, Donald E. Brown, Kiana Jafari Meimandi, and Laura E. Barnes. RMDL: random multimodel deep learning for classification. *CoRR*, abs/1805.01890, 2018.

Yann LeCun, Corinna Cortes, and Chris Burges. MNIST handwritten digit database, 1998. URL http://yann.lecun.com/exdb/mnist/.

Changliu Liu, Tomer Arnon, Christopher Lazarus, Clark Barrett, and Mykel J. Kochenderfer. Algorithms for verifying deep neural networks. *CoRR*, abs/1903.06758, 2019. URL http://arxiv.org/abs/1903.06758.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations (ICLR)*, 2017.

Pei-Hsuan Lu, Pin-Yu Chen, Kang-Cheng Chen, and Chia-Mu Yu. On the limitation of MagNet defense against L_1-based adversarial examples. In *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN)*, pp. 200–214. IEEE Computer Society, 2018. doi: 10.1109/DSN-W.2018.00065. URL http://doi.ieeecomputersociety.org/10.1109/DSN-W.2018.00065.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017. URL http://arxiv.org/abs/1706.06083.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1, 2014. URL http://arxiv.org/abs/1411.1784.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In Doina Precup and Yee Whye Teh (eds.), *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2642–2651. PMLR, 2017. URL http://proceedings.mlr.press/v70/odena17a.html.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge J. Belongie. Generative adversarial perturbations. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 4422–4431. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00465. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Poursaeed_Generative_Adversarial_Perturbations_CVPR_2018_paper.html.

Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska. Global robustness evaluation of deep neural networks with provable guarantees for L0 norm. *CoRR*, abs/1804.05805, 2018. URL http://arxiv.org/abs/1804.05805.

Tim Salimans, Ian J Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In Daniel D Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2226–2234, 2016. URL http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.

Lukas Schott, Jonas Rauber, Wieland Brendel, and Matthias Bethge. Towards the first adversarially robust neural network model on MNIST. *CoRR*, abs/1805.09190, 2018. URL http://arxiv.org/abs/1805.09190.

Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *CoRR*, abs/1809.02104, 2018. URL http://arxiv.org/abs/1809.02104.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. A general framework for adversarial examples with objectives. *ACM Trans. Priv. Secur.*, 22(3):16:1–16:30, 2019. doi: 10.1145/3317611. URL https://doi.org/10.1145/3317611.

Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206 of *Lecture Notes in Computer Science*, pp. 218–234. Springer, 2018. ISBN 978-3-030-01215-1. doi: 10.1007/978-3-030-01216-8_14. URL https://doi.org/10.1007/978-3-030-01216-8_14.

Qing Song, Yingqi Wu, and Lu Yang. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. *CoRR*, abs/1811.12026, 2018a. URL http://arxiv.org/abs/1811.12026.

Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pp. 8322–8333, 2018b. URL https://arxiv.org/abs/1805.07894.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *International Conference on Learning Representations (ICLR)*, 2014. URL http://arxiv.org/abs/1312.6199.

Li Wan, Matthew D. Zeiler, Sixin Zhang, Yann LeCun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning (ICML)*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1058–1066. PMLR, 2013. URL http://proceedings.mlr.press/v28/wan13.html.

Shiqi Wang, Yizheng Chen, Ahmed Abdou, and Suman Jana. Mixtrain: Scalable training of formally robust neural networks. *CoRR*, abs/1811.02625, 2018.

Xiaosen Wang, Kun He, and John E. Hopcroft. AT-GAN: A generative attack model for adversarial transferring on generative adversarial nets. *CoRR*, abs/1904.07793, 2019. URL http://arxiv.org/abs/1904.07793.

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Jennifer G Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *JMLR Workshop and Conference Proceedings*, pp. 5283–5292. JMLR.org, 2018. URL http://proceedings.mlr.press/v80/wong18a.html.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In Jérôme Lang (ed.), *International Joint Conferences on Artificial Intelligence (IJCAI)*, pp. 3905–3911. ijcai.org, 2018. ISBN 978-0-9992411-2-7. doi: 10.24963/ijcai.2018/543. URL https://doi.org/10.24963/ijcai.2018/543.

Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil Jain. Adversarial attacks and defenses in images, graphs and text: A review. *CoRR*, abs/1909.08072, 2019. URL https://arxiv.org/abs/1909.08072.

Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7354–7363. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhang19d.html.

A SAMPLES OF GENERATED ADVERSARIAL EXAMPLES ON IMAGENET

Randomly selected unrestricted adversarial samples generated using BigGAN (Brock et al., 2019).

Figure 8: Images classified as ‘tabby cat’ and their intended true class.

Figure 9: Images classified as ‘slug’ and their intended true class.

Figure 10: Images classified as ‘orange’ and their intended true class.

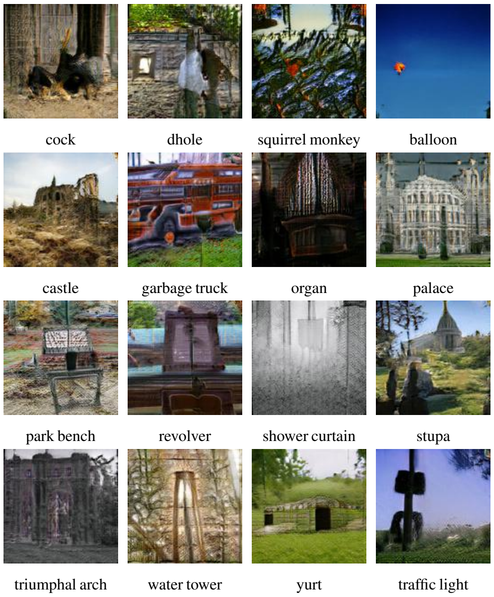
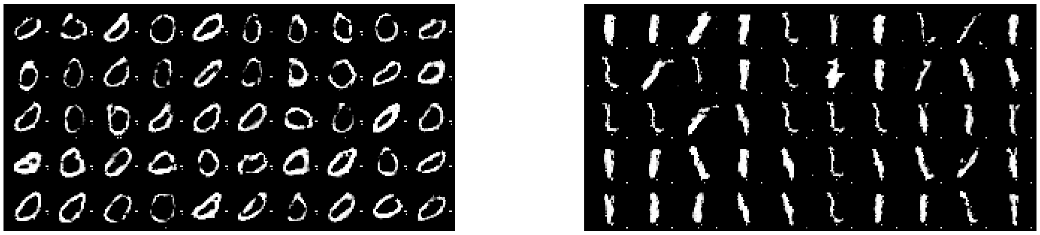


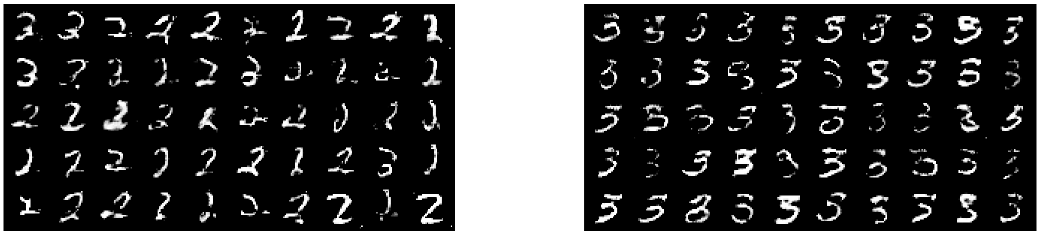
Figure 11: Images classified as ‘church’ and their intended true class.

B SAMPLES OF GENERATED UNRESTRICTED ADVERSARIAL EXAMPLES

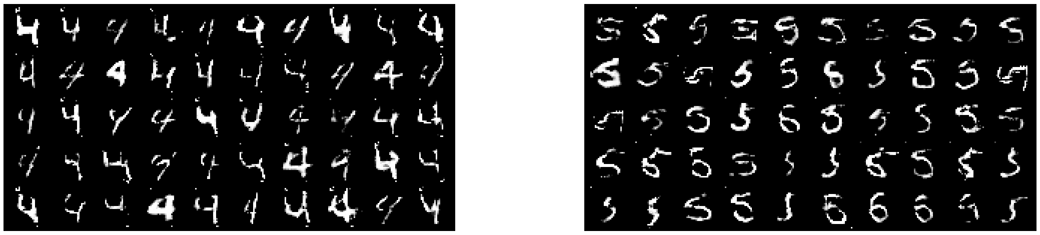
These samples were generated by finetuned GANs targeting Wong & Kolter (2018) locally-robust classifier.



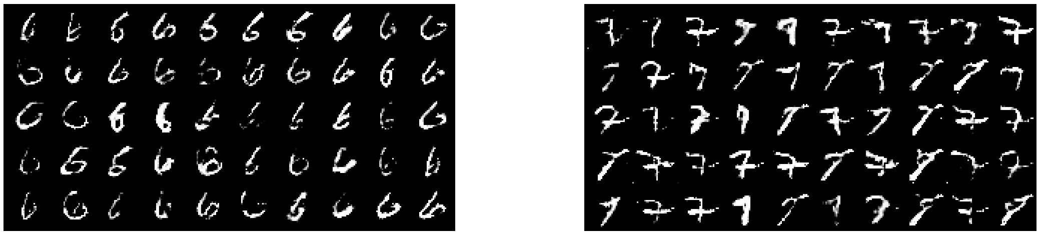
(a) Intended true label ‘0’. (b) Intended true label ‘1’.



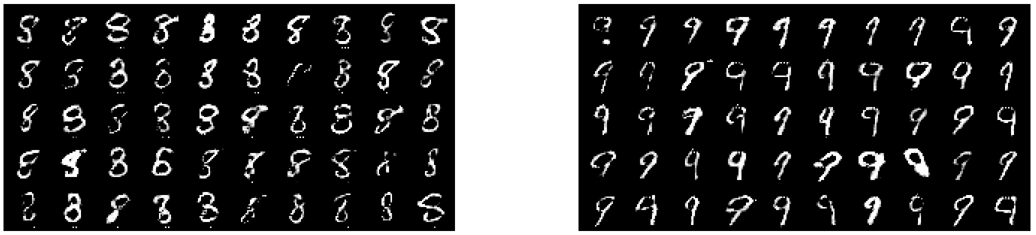
(c) Intended true label ‘2’. (d) Intended true label ‘3’.



(e) Intended true label ‘4’. (f) Intended true label ‘5’.



(g) Intended true label ‘6’. (h) Intended true label ‘7’.



(i) Intended true label ‘8’. (j) Intended true label ‘9’.

Figure 12: Examples of expected-adversarial images generated by a GAN finetuned to perform an untargeted attack on Wong and Kolter’s locally-robust classifier network.

C TARGETED CLASSIFIERS

All targeted classifiers (other than ‘simple fully-connected’) are provably robust to adversarial perturbations in the sense that there is guaranteed to be no adversarial input within a distance ϵ of $p\%$ of test inputs under the l_∞ norm.

Table 3: Descriptions of and references to target classfiers used.

Our Name	Abbreviation	ϵ	p	Architecture
Wong & Kolter (2018)	W&K	0.1	94.2	2 convolutional layers followed by 2 dense layers
MixTrain (Wang et al., 2018) Model A	MT-A	0.1	97.1	‘MNIST_small’: 2 convolutional layers followed by 1 dense layer
MixTrain (Wang et al., 2018) Model B	MT-B	0.3	60.1	‘MNIST_small’: 2 convolutional layers followed by 1 dense layer
MixTrain (Wang et al., 2018) Model C	MT-C	0.1	96.4	‘MNIST_large’: 4 convolutional layers followed by 2 dense layers
MixTrain (Wang et al., 2018) Model D	MT-D	0.3	58.4	‘MNIST_large’: 4 convolutional layers followed by 2 dense layers
Simple Fully-Connected	Simple	N/A	N/A	Three fully-connected layers of size 256, 128 and 32 with LeakyReLU activations

D ADVERSARIAL TRAINING EXPERIMENT

The classifier trained during adversarial training (both the architecture and hyperparameters) is the one used in Madry et al. (2017), and in particular from their associated MNIST Adversarial Examples Challenge.

For the experiments with our own model, we first pre-train the generator. We then continue in ‘training rounds’. First, we fine-tune against the classifier for 5000 gradient steps, using the hyperparameters from Table 6, but with an attack rate of 0.4. Next, we produce 80,000 attacked training examples (using an untargeted attack), which are added to the pool of all examples generated so far. Then, the classifier is trained on the entirety of the pool of samples 30 times, with a batch size of 128. Once a training round is completed we start again, resetting the GAN to how it was *before* the fine-tuning.

For the experiments with Song et al.’s model, we run 300 training epochs for the Madry et al. classifier, with a training batch size of 64. At each step, the training data is produced by Song et. al.’s model. We use their code and the hyperparameters they provide for untargeted attacks in Table 4 of their appendix.

E NEURAL NETWORK ARCHITECTURES AND HYPERPARAMETERS

The WGAN-GP (Gulrajani et al., 2017) and ACGAN (Odena et al., 2017) architectures were the starting points for the design of these neural networks. Only a small amount of manual hyperparameter tuning was performed.

The discriminator network is a combination of a conditional WGAN-GP critic, which learns an approximation of the Wasserstein distance between the generated and training-set conditional distributions, and an auxiliary classifier, which predicts the likelihood of the possible values of $h(x)$. We combined these two architectures in an attempt to strengthen the gradient provided to the generator, helping to generate data which are both realistic and for which the true (i.e., human-judged) labels match the intended true labels. The critic is given the true label of the data $h(x)$ to improve its training, but the auxiliary classifier must not have access to this information since its purpose is to predict it. We therefore split the discriminator d into three sub-networks. Network $d_0: X \to \mathbb{R}^i$ effectively preprocesses the input, passing an intermediate representation to the critic network $d_1: \mathbb{R}^i \times Y \to \mathbb{R}$ and the auxiliary classifier network $d_2: \mathbb{R}^i \to \mathbb{R}^{|Y|}$. In our experiments, both d_1 and d_2 were single fully-connected layers of the appropriate dimension. The loss functions from the WGAN-GP and ACGAN algorithms are simply summed.

Table 4: Architecture for generator network, g.

Layer Type	Kernel	Strides	Feature Maps	Batch Norm.	Dropout	Activation
Fully-Connected	N/A	N/A	64	No	0	ReLU
Transposed Convolution	5×5	2×2	32	Yes	0.35	LeakyReLU
Transposed Convolution	5×5	2×2	8	Yes	0.35	LeakyReLU
Transposed Convolution	5×5	2×2	4	Yes	0.35	LeakyReLU
Fully-Connected	N/A	N/A	784	No	0	Tanh

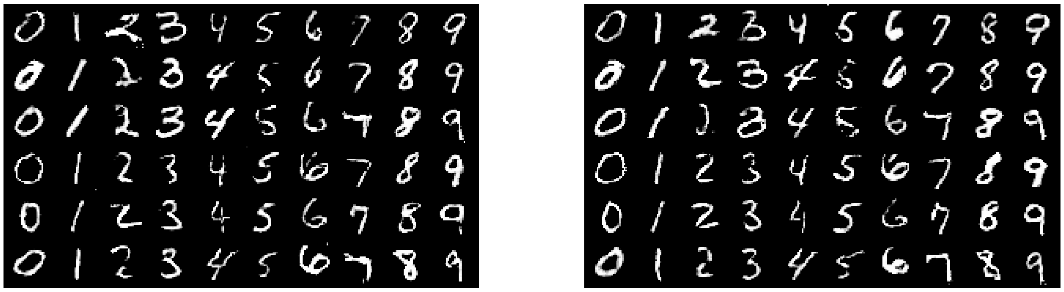
Table 5: Architecture for discriminator subnetwork, d_0.

Layer Type	Kernel	Strides	Feature Maps	Batch Norm.	Dropout	Activation Function
Convolution	3×3	2×2	8	No	0.2	LeakyReLU
Convolution	3×3	1×1	16	No	0.2	LeakyReLU
Convolution	3×3	2×2	32	No	0.2	LeakyReLU
Convolution	3×3	1×1	64	No	0.2	LeakyReLU
Convolution	3×3	2×2	128	No	0.2	LeakyReLU
Convolution	3×3	1×1	256	No	0.2	LeakyReLU

Table 6: Hyperparameters for all networks.

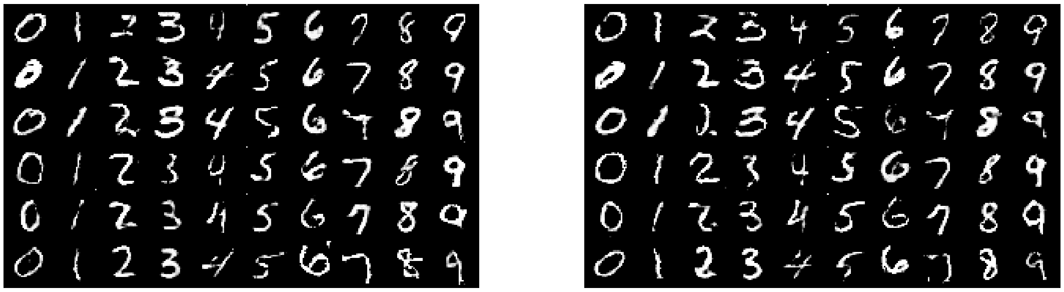
Hyperparameter	Value
Attack rate	$\mu = 0.1$
Learning rate	$\alpha = 0.000005$
Adam betas	$\beta_1 = 0.6, \beta_2 = 0.999$
Leaky ReLU slope	0.2
Minibatch size	100
Dimensionality of latent space	128
Weight initialisation	Normally distributed as described by He et al. (2015)
Coefficient of gradient penalty loss term	$\lambda = 10$

F VISUAL EFFECT OF ADVERSARIAL FINETUNING



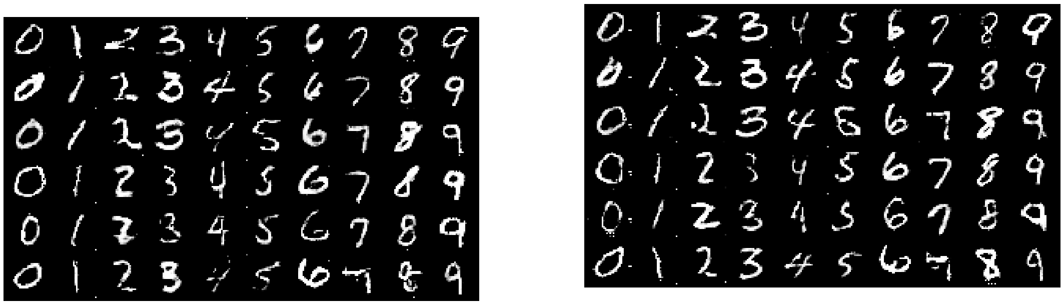
(a) Samples from pretrained generator.

(b) After 5,000 iterations of finetuning.



(c) After 10,000 iterations of finetuning.

(d) After 20,000 iterations of finetuning.



(e) After 30,000 iterations of finetuning.

(f) After 45,000 iterations of finetuning. We ended finetuning at this stage.

Figure 13: A sequence of images tracking the output of the generator network for one fixed random sample in latent space as adversarial finetuning takes place. Five samples are given for each intended true label. The finetuning is an untargeted attack against Wong & Kolter (2018) locally-robust network.

G CAN OUR METHOD BE USED WITH ONLY BLACK-BOX ACCESS?

Perturbation-based adversarial examples typically somewhat generalise between models (Szegedy et al., 2014; Liu et al., 2017). That is, inputs crafted using white-box access to fool one model often fool a different model. This means that black-box attacks are possible, if the attacker has a different trained model for the same task. To evaluate whether our method could be used in the same way, we generated about 20,000 untargeted unrestricted adversarial inputs for each target classifier, and measured the misclassification rates on this set for the other models. The high variance of the results, shown in Table 7, suggests that successful transfer may depend more on the networks in question than on our generation algorithm.

Table 7: The percentage of adversarial examples targeting each classifier which are also adversarial for the others. See Appendix C for descriptions of the classifiers.

		To					
		W&K	MT-A	MT-B	MT-C	MT-D	Simple
From	W&K		20.2	18.4	9.0	60.7	16.8
	MT-A	19.5		14.1	13.3	55.2	4.7
	MT-B	5.2	4.8		1.6	57.8	2.6
	MT-C	25.8	47.6	13.9		67.8	12.1
	MT-D	5.9	7.3	9.4	4.3		1.7
	Simple	2.7	2.6	2.6	1.3	48.0	

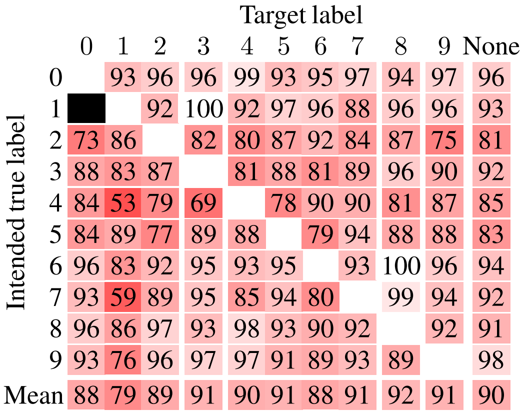
H MEASURES TAKEN TO ENSURE MTURK DATA QUALITY

The evaluation of our method relies entirely on the quality of the data provided by the MTurk workers. We therefore took a number of measures to ensure that participants understood the instructions and completed the tasks diligently:

- Only workers with good track records were permitted to participate.
- The instructions specified that particular answers should be given to specified questions to prove that the instructions had been read carefully. Approximately 10% of work was rejected for failing this check.
- For the image labelling tasks, some images with known labels were included to check that the right labels were being given. Reassuringly, almost no work was rejected for failing this check.
- For the identification of the generated images, a bonus nearly doubling the pay per image was given for each correctly-identified image, providing an extra incentive to try hard.
- To provide a disincentive to high-speed random clicking, a minimum time spent answering each question was enforced.
- If more than 1% of questions were left unanswered, we interpreted this as a sign of carelessness and did not use any of the data from that task.

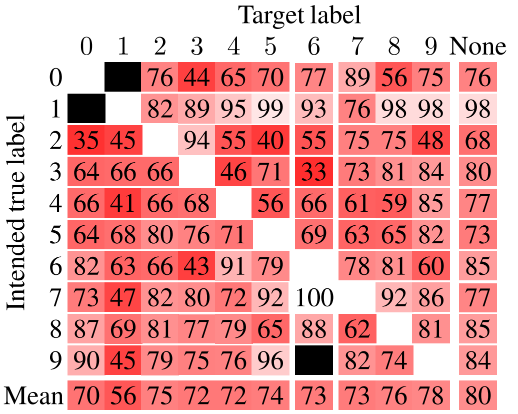
I RESULTS OF EXPERIMENTS FOR NON-ROBUST TARGET NETWORK

These results are targeting a simple convolutional neural network with LeakyReLU activations and three hidden layers of size 256, 128 and 32, trained until convergence.



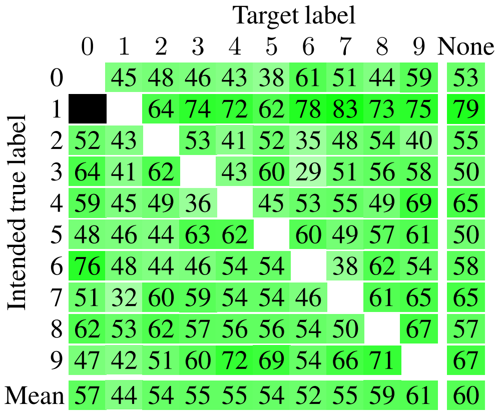
	Target label										
Intended true label	0	1	2	3	4	5	6	7	8	9	None
0		93	96	96	99	93	95	97	94	97	96
1	■		92	100	92	97	96	88	96	96	93
2	73	86		82	80	87	92	84	87	75	81
3	88	83	87		81	88	81	89	96	90	92
4	84	53	79	69		78	90	90	81	87	85
5	84	89	77	89	88		79	94	88	88	83
6	96	83	92	95	93	95		93	100	96	94
7	93	59	89	95	85	94	80		99	94	92
8	96	86	97	93	98	93	90	92		92	91
9	93	76	96	97	97	91	89	93	89		98
Mean	88	79	89	91	90	91	88	91	92	91	90

Figure 14: The success rates of the adversarial attacks by finetuned GANs. More precisely, of generated images for which the computed label output by the classifier matches the target label, the percentage which are truly adversarial (in the sense that the true label of the image matches the intended true label passed to the generator network) is reported.



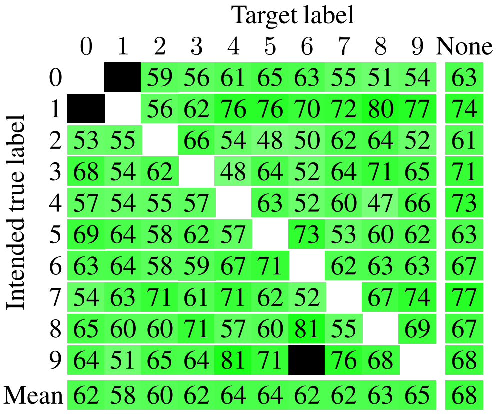
	Target label										
Intended true label	0	1	2	3	4	5	6	7	8	9	None
0		■	76	44	65	70	77	89	56	75	76
1	■		82	89	95	99	93	76	98	98	98
2	35	45		94	55	40	55	75	75	48	68
3	64	66	66		46	71	33	73	81	84	80
4	66	41	66	68		56	66	61	59	85	77
5	64	68	80	76	71		69	63	65	82	73
6	82	63	66	43	91	79		78	81	60	85
7	73	47	82	80	72	92	100		92	86	77
8	87	69	81	77	79	65	88	62		81	85
9	90	45	79	75	76	96	■	82	74		84
Mean	70	56	75	72	72	74	73	73	76	78	80

Figure 15: The success rates of the adversarial attacks by a pretrained but not finetuned GAN. More precisely, of generated images for which the computed label output by the classifier matches the target label, the percentage which are truly adversarial (in the sense that the true label of the image matches the intended true label passed to the generator network) is reported.



	Target label										
Intended true label	0	1	2	3	4	5	6	7	8	9	None
0		45	48	46	43	38	61	51	44	59	53
1	■		64	74	72	62	78	83	73	75	79
2	52	43		53	41	52	35	48	54	40	55
3	64	41	62		43	60	29	51	56	58	50
4	59	45	49	36		45	53	55	49	69	65
5	48	46	44	63	62		60	49	57	61	50
6	76	48	44	46	54	54		38	62	54	58
7	51	32	60	59	54	54	46		61	65	65
8	62	53	62	57	56	56	54	50		67	57
9	47	42	51	60	72	69	54	66	71		67
Mean	57	44	54	55	55	54	52	55	59	61	60

Figure 16: Measures of how realistic the adversarial images generated by finetuned GANs are. More precisely, the proportion of generated inputs for which the classified label matches the target label which were not identified as being generated when placed amongst nine images from the training dataset. If the generated images were completely realistic, the expected result would be 90.

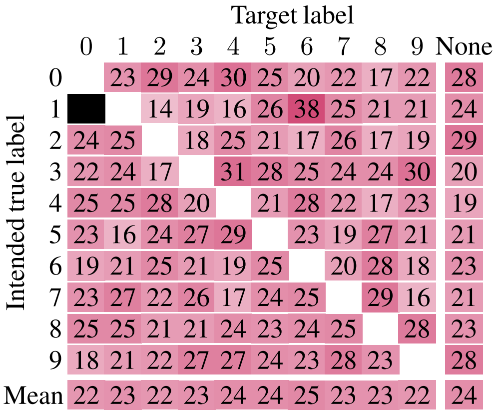


	Target label										
Intended true label	0	1	2	3	4	5	6	7	8	9	None
0		■	59	56	61	65	63	55	51	54	63
1	■		56	62	76	76	70	72	80	77	74
2	53	55		66	54	48	50	62	64	52	61
3	68	54	62		48	64	52	64	71	65	71
4	57	54	55	57		63	52	60	47	66	73
5	69	64	58	62	57		73	53	60	62	63
6	63	64	58	59	67	71		62	63	63	67
7	54	63	71	61	71	62	52		67	74	77
8	65	60	60	71	57	60	81	55		69	67
9	64	51	65	64	81	71	■	76	68		68
Mean	62	58	60	62	64	64	62	62	63	65	68

Figure 17: Measures of how realistic the adversarial images generated by a pretrained but not finetuned GAN are. More precisely, the proportion of generated inputs for which the classified label matches the target label which were not identified as being generated when placed amongst nine images from the training dataset. If the generated images were completely realistic, the expected result would be 90.

J RESULTS OF SIDE-BY-SIDE COMPARISON EXPERIMENTS

Each figure shows the number of human judgements out of 100 which correctly identified the unrestricted adversarial input in a side-by-side comparison with an image drawn from the dataset. If the generated images were completely realistic, the expected result would be 50.

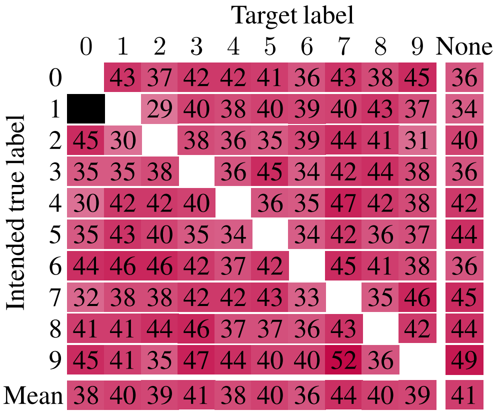


Target label

Intended true label

	0	1	2	3	4	5	6	7	8	9	None
0		23	29	24	30	25	20	22	17	22	28
1	■		14	19	16	26	38	25	21	21	24
2	24	25		18	25	21	17	26	17	19	29
3	22	24	17		31	28	25	24	24	30	20
4	25	25	28	20		21	28	22	17	23	19
5	23	16	24	27	29		23	19	27	21	21
6	19	21	25	21	19	25		20	28	18	23
7	23	27	22	26	17	24	25		29	16	21
8	25	25	21	21	24	23	24	25		28	23
9	18	21	22	27	27	24	23	28	23		28
Mean	22	23	22	23	24	24	25	23	23	22	24

Figure 18: Results against the locally-robust neural network generated by finetuned GANs.

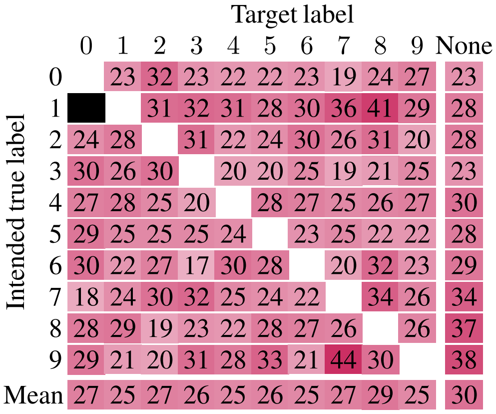


Target label

Intended true label

	0	1	2	3	4	5	6	7	8	9	None
0		43	37	42	42	41	36	43	38	45	36
1	■		29	40	38	40	39	40	43	37	34
2	45	30		38	36	35	39	44	41	31	40
3	35	35	38		36	45	34	42	44	38	36
4	30	42	42	40		36	35	47	42	38	42
5	35	43	40	35	34		34	42	36	37	44
6	44	46	46	42	37	42		45	41	38	36
7	32	38	38	42	42	43	33		35	46	45
8	41	41	44	46	37	37	36	43		42	44
9	45	41	35	47	44	40	40	52	36		49
Mean	38	40	39	41	38	40	36	44	40	39	41

Figure 19: Results against the locally-robust neural network generated by a pretrained but not finetuned GAN.

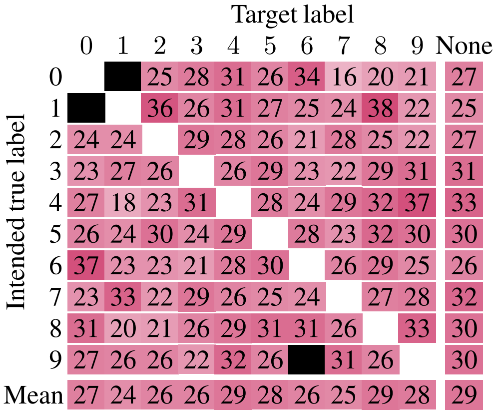


Target label

Intended true label

	0	1	2	3	4	5	6	7	8	9	None
0		23	32	23	22	22	23	19	24	27	23
1	■		31	32	31	28	30	36	41	29	28
2	24	28		31	22	24	30	26	31	20	28
3	30	26	30		20	20	25	19	21	25	23
4	27	28	25	20		28	27	25	26	27	30
5	29	25	25	25	24		23	25	22	22	28
6	30	22	27	17	30	28		20	32	23	29
7	18	24	30	32	25	24	22		34	26	34
8	28	29	19	23	22	28	27	26		26	37
9	29	21	20	31	28	33	21	44	30		38
Mean	27	25	27	26	25	26	25	27	29	25	30

Figure 20: Results against an ordinary neural network generated by finetuned GANs.



Target label

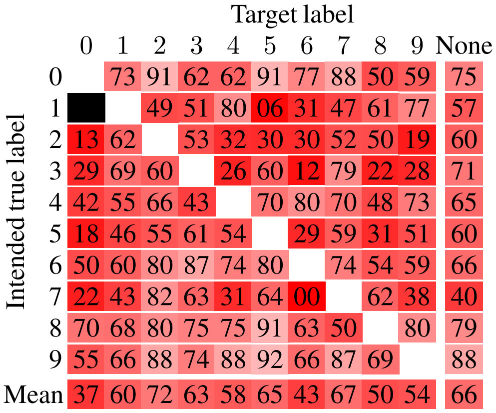
Intended true label

	0	1	2	3	4	5	6	7	8	9	None
0		■	25	28	31	26	34	16	20	21	27
1	■		36	26	31	27	25	24	38	22	25
2	24	24		29	28	26	21	28	25	22	27
3	23	27	26		26	29	23	22	29	31	31
4	27	18	23	31		28	24	29	32	37	33
5	26	24	30	24	29		28	23	32	30	30
6	37	23	23	21	28	30		26	29	25	26
7	23	33	22	29	26	25	24		27	28	32
8	31	20	21	26	29	31	31	26		33	30
9	27	26	26	22	32	26	■	31	26		30
Mean	27	24	26	26	29	28	26	25	29	28	29

Figure 21: Results against an ordinary neural network generated by a pretrained but not finetuned GAN.

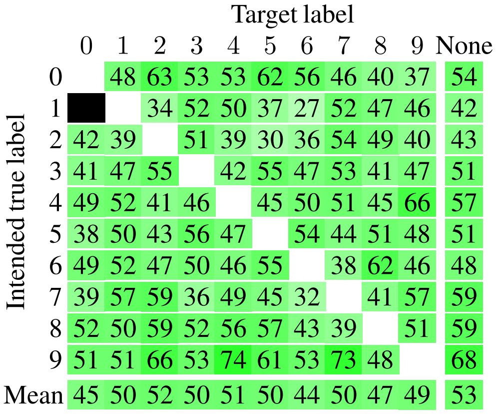
K RESULTS OF EXPERIMENTS FOR PRETRAINED BASELINE

These results are for data generated by a GAN which has been pretrained but not adversarially finetuned, targeting Wong & Kolter (2018) provably-robust network.



	Target label										
Intended true label	0	1	2	3	4	5	6	7	8	9	None
0		73	91	62	62	91	77	88	50	59	75
1	■		49	51	80	06	31	47	61	77	57
2	13	62		53	32	30	30	52	50	19	60
3	29	69	60		26	60	12	79	22	28	71
4	42	55	66	43		70	80	70	48	73	65
5	18	46	55	61	54		29	59	31	51	60
6	50	60	80	87	74	80		74	54	59	66
7	22	43	82	63	31	64	00		62	38	40
8	70	68	80	75	75	91	63	50		80	79
9	55	66	88	74	88	92	66	87	69		88
Mean	37	60	72	63	58	65	43	67	50	54	66

Figure 22: The success rates of the adversarial attacks by a pretrained but not finetuned GAN. More precisely, of generated images for which the computed label output by the classifier matches the target label, the percentage which are truly adversarial (in the sense that the true label of the image matches the intended true label passed to the generator network) is reported.



	Target label										
Intended true label	0	1	2	3	4	5	6	7	8	9	None
0		48	63	53	53	62	56	46	40	37	54
1	■		34	52	50	37	27	52	47	46	42
2	42	39		51	39	30	36	54	49	40	43
3	41	47	55		42	55	47	53	41	47	51
4	49	52	41	46		45	50	51	45	66	57
5	38	50	43	56	47		54	44	51	48	51
6	49	52	47	50	46	55		38	62	46	48
7	39	57	59	36	49	45	32		41	57	59
8	52	50	59	52	56	57	43	39		51	59
9	51	51	66	53	74	61	53	73	48		68
Mean	45	50	52	50	51	50	44	50	47	49	53

Figure 23: Measures of how realistic the adversarial images generated by a pretrained but not finetuned GAN are. More precisely, the proportion of generated inputs for which the classified label matches the target label which were not identified as being generated when placed amongst nine images from the training dataset. If the generated images were completely realistic, the expected result would be 90.