Defenses Against Self-Adversarial Attacks in Cycle-Consistent GANs

Jonathan Chu, Bryan Lee, Yi Lin Wang, Catherine Zeng *
Harvard University

Abstract

GANs have recently gained significant popularity as a state-of-the-art technique for unsupervised image generation. However, cycle-consistent GANs (CycleGANs) have been shown to be easily susceptible to adversarial attacks. We reproduce and extend previous work in adversarial attacks against cycle-consistent GANs from [2], successfully demonstrating the robustness of the authors' proposed adversarial attack defense mechanisms. Additionally, we contribute new performance metrics for adversarial defenses, illustrate the weakness of the defense mechanisms in WGANs, and contribute a novel defense mechanism, the 4D defense.

1 Introduction

In this project, we verify and extend state-of-the-art techniques in the field of unsupervised image-to-image translation. In such a setting where we seek to learn a mapping between two image domains, given *unpaired* data from both domains, recent work has demonstrated that CycleGANs, bidirectional pairs of GANs supplemented with an additional cycle-consistency loss, have great capacity to learn the desired human-intuitive mapping – one which preserves the most visual semantic information through translation. However, recent literature also exposes CycleGANs' vulnerability to *self-adversarial attacks*, an observation of CycleGANs' tendency to embed hidden information in its image translations in order to assist the minimization of reconstruction loss. While helpful for the reconstruction objective, this behavior causes the model to produce less stable translations when fed inputs with small, noisy variations.

In particular, Bashkirova et al. [2] introduce two defense mechanisms against self-adversarial attacks for CycleGANs. We first reproduce the experiments in their paper on a Google Maps data set, and we additionally conduct a hyperparameter search to determine the robustness of the paper's techniques. We then extend the results in three ways: 1) we combine their approach with another GAN objective function, 2) we propose a new metric for reconstruction honesty, and 3) we propose and test a new defense technique.[1]

2 Background and Related Work

In 2014, Goodfellow et al. [5] introduced generative adversarial networks (GANs), which are a type of network structure consisting of a generator and discriminator. Since then, GANs have been vigorously studied and employed for image generation, and found to be able to generate high quality images in many domains [13] [6] [4] [12]. For mapping $G : X \rightarrow Y$ that attempts to generate images

*{jonathanchu,bryanlee,yilinwang,catherinezeng}@college.harvard.edu

[1] Our code can be found at https://github.com/Bryanlee99/pix2pix_cyclegan_guess_noise.

that look like real ones from domain Y and discriminator D_Y that tries to distinguish between $G(x)$ instances and real instances $y \in Y$, GANs typically minimize the adversarial loss function

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))]. \quad (1)$$

A similar loss is used for the reverse mapping $F : Y \to X$, given by $\mathcal{L}_{\text{GAN}}(F, D_X, Y, X)$ [14].

Unsupervised image-to-image translation refers to the problem of mapping images from one domain A to another domain B without ground truth labels that provide correspondence between examples in the domains. This problem has been previously addressed by conditional GANs [9] [7]. Recently, Zhu et al. [14] proposed a new structure called a CycleGAN for unsupervised image-to-image translation. A CycleGAN consists of two discriminator and two generator networks. One generator is trained to produce realistic translations of images in domain A to images in domain B, and the other generator is trained to produce realistic translations of images in domain B to images in domain A. The discriminators are trained to distinguish between generated and real images in each domain.

In particular, the strategy employed by a CycleGAN to generate realistic images is through enforcing a so-called cycle consistency property. Suppose the goal is to learn a realistic mapping G of images in domain A to translations in domain B. Then ideally, with an excellent G, we might expect that we can translate backwards. That is, we would want a mapping of images F from domain B to translations in domain A such that for each image x in domain A, its reconstruction satisfies $F(G(x)) \approx x$. In order to facilitate this cycle consistency property, we train the generators and discriminators to learn mappings F and G subject to a loss based on a pixelwise distance between input image and reconstruction given by

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]. \quad (2)$$

Enforcing a small error between true input images and their reconstructions often ensures that the translated images preserve the semantics of the input image, like shape and position of objects. Combining the adversarial loss and cyclic loss, our full objective becomes

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(D, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda_A \mathcal{L}_{\text{cyc}}(G, F), \quad (3)$$

where λ_A weights the importance of the two objectives [14].

However, Chu et al. [3] found that CycleGANs sometimes suffer from a major flaw; the reconstruction loss can cause the generator network to hide information necessary to faithfully reconstruct the input image inside tiny perturbations of the translated image. If domain A is richer than B, then for some image y in domain B, there may be multiple correct reconstructions in domain A. But a CycleGAN may attempt to hide information about aspects of image x into its translation of x, that are nearly invisible to the human eye, but which it nonetheless depends on for reconstruction. Thus we can essentially 'hack' a CycleGAN that achieves the reconstruction we want regardless of what the translation looks like, since we can always hide information in the translation. Because the translation does not truly reflect semantic details in the input image, the hiding of information in this context has been called a self-adversarial attack [2].

The main paper we are reproducing and extending [2] has proposed two defense techniques to defend against self-adversarial attacks. One approach is to add perturbations to translated images before undergoing reconstruction, so that with high probability, any invisible embedded information to help with reconstruction is destroyed and the CycleGAN never learns to rely on these features to help it reconstruct. The modified reconstruction loss in this method looks like

$$\mathcal{L}_{rec}^{noisy} = \|F(G(x) + \Delta(\theta_n)) - x\|_1,$$

where $\Delta(\theta_n)$ is a high-frequency perturbation function with parameters θ_n.

Another approach is the addition of a guess discriminator. A discriminator in a CycleGAN has difficulty detecting self-adversarial attacks because it doesn't see ground truth examples (real and fake) of the same images. This approach approximates giving the CycleGAN fake (adversarial) examples by using the reconstruction output as a fake example. We have a new discrimator called a guess discriminator D_{guess} that receives both a real input image and a reconstruction in random order,

and guesses which is fake. D_{guess} tries to minimize its error while the generator tries to produce images to maximize the error of D_{guess}.

The paper above uses metrics of noise sensitivity and reconstruction honesty to assess the effectiveness of defenses against such adversarial attacks. The authors train a CycleGAN using these defenses and find that these defense strategies are effective at reducing hidden information according to their metrics.

3 Approach

3.1 Infrastructure

We use a Google Maps data set that contains roughly 3000 Google Maps photos. Domain *A* contains real satellite images, and domain *B* contains "segmented" maps images that depict buildings, roads, water, and lawns. For this dataset, the maps domain is not truly a set of segmentation maps with a small set of possible values for the pixels; it consists of actual Google Maps photos that have blurry edges. Therefore, to enforce a reasonable classification structure for the calculation of our accuracy metrics, we perform even quantization (discussed in the section 3.3 below). Bashkirova et al. [2] also tested their defenses on the GTA [10] and SynAction [11] datasets, but we decided not to use them. The GTA dataset is very large (exceeding 50 GB), and it would have taken an enormous amount of time to train our models with the compute available to us. The SynAction dataset used by the authors is not publicly available, although the authors plan to release it with a future paper.

Conducting hyperparameter searches and testing new defense mechanisms was very compute intensive and expensive. The models were trained using six Tesla V100 GPUs on AWS. Overall, GPU costs for this project totalled $400.

3.2 Reproductions

3.2.1 Models

We reproduced the final models developed in [2] and conducted supplementary hyperparameter searches to determine the robustness of the authors' algorithms. The parameters of the authors' baseline CycleGAN algorithm are described in the appendix. In particular, we test the following baseline algorithms using the authors' selected hyperparameters:

1. CycleGAN: $\lambda_A = 10$, $\lambda_B = 10$
2. CycleGAN + noise: $\sigma = 0.06$, $\lambda_A = 10$, $\lambda_B = 10$
3. CycleGAN + guess discriminator: $\lambda_{guess} = 1$, $\lambda_A = 1$, $\lambda_B = 2$
4. CycleGAN + noise + guess discriminator: $\lambda_{guess} = 1$, $\sigma = 0.06$, $\lambda_A = 1$, $\lambda_B = 2$

Specifically, λ_A and λ_B are the cyclic consistency losses for $A - B - A$ (A2B2A) and $B - A - B$ (B2A2B), respectively, σ is the standard deviation of the additive gaussian noise, and λ_{guess} is the weight of the cycle loss for $A2A$ applied to the guess discriminator.

Furthermore, we tested the baseline algorithms with multiple sets of hyperparameters. We list the categories of hyperparameters tuned and the intuitive rationale.

1. ADAM Momentum (M), Learning Rate (L_r), and Learning Rate Policy (L_p): Cycle-GANs are difficult to train robustly because the simultaneous training of the generator and discriminator is unstable. To investigate the instability in Cycle-GAN training, we modified trained with $L_r = 0.0002, 0.0004$, $L_p =$ linear, step, cosine, and $M = 0.5, 0.6$.
2. λ_A, λ_B: The relative values of λ_A and λ_B are important when translating between domains of different entropies. Specifically, providing a high penalty to one-to-many translations (i.e. $\lambda_A < \lambda_B$, where domain A has a higher entropy than domain B) encourages the Cycle-GAN to penalize incorrect translations from the domain of lower entropy (i.e. the Maps 2D view) to that of higher entropy (i.e. the colorized Maps 3D view), which are more likely than the reverse.
3. Noise, σ: Training with noise makes the algorithm more robust to adversarial Gaussian noise. Given the authors' default parameter was $\sigma = 0.06$, we tested $\sigma = 0.04, 0.06, 0.10$.

The full list of models and their hyperparameters are listed in tables 1 and 2 in the appendix.

3.2.2 Defense Mechanisms

For our reproduction, we tested CycleGAN models using the authors' two proposed defense mechanisms across four CycleGAN models. These models include a baseline CycleGAN model, a model with additive noise, a model with the guess discriminator, and a model with both additive noise and the guess discriminator. The hyperparameters used were identical to those used by the authors and are listed in section 3.2.1 summarizing the model design. Sample image translations and reconstructions using these four models are shown in the section 4 on results.

3.2.3 Metrics

The authors of recent work propose three classes of metrics to evaluate different components of the models: translation quality, reconstruction honesty, and sensitivity to noise. We specify our interpretation and implementation of these metrics.

Before examining the metrics themselves, an important technique we used for calculating our metrics was the quantization operation $\lfloor * \rfloor$. Given some input image Z, the idea of the quantization $\lfloor Z \rfloor$ is to produce a new image where the each pixel of Z is rounded off to nearest pixel value from a set S of pixel values. We experimented with two such sets S:

1. **Even quantization**. Here, $S_q = \{j * \frac{255}{q} | j \in \mathbb{N}, j \leq q\}$. We divide the RGB spectrum into q even segments, and round each pixel to the nearest subdivision. Thus, $|S_q| = q + 1$. For our experiments, we chose $q = 20$.
2. **Palette quantization**. Here, we assume there is some finite set P (the "palette") of pixel values present in the image domain of Z, and $S = P$. For example, if the domain of Z is semantic segmentation maps, we might expect $|S| \sim 10^1$, where different colors label different classes of objects (ground, car, tree, sky, etc.).

Quantization of translated images was used to enforce the classification nature of the map image domain–they are semantic segmentation masks. In implementing palette quantization, we faced the additional difficulty that the images in our segmented image domain, Google maps images, were not "cleanly" semantically segmented; even when converted to grayscale, almost all images had > 200 unique pixel values. Google maps images, while expressing segmentations that are clear to humans, are not perfect segmentation masks in themselves. Therefore, we manually identified the 8 most popular pixel color values which constituted our set S.

Ultimately, we decided to use the even quantization strategy due to computational efficiency. Implementation-wise, the quantization algorithm is $O(n^2|S|)$ when nearest-value rounding is determined according to the Euclidean distance between RGB values in $|S|$, or perhaps $O(n^2 \log |S| + |S| \log |S|) \sim O(n^2 \log |S|)$ when $|S|$ is ordered in the grayscale case (n is the side length of the image). However, even with $|S| = 8$, this proved too costly when executed in a Python loop, along with latency associated with the writing to PyTorch tensors to and from various devices especially when executed on Google Colab (GPU, mounted Drive, etc.) The even quantization, however, was easily implementable using native rounding methods from PyTorch and/or NumPy, which take advantage of C/C++ loops and other optimizations.

Translation Accuracy. In order to evaluate the quality of the learned mapping $G_B : X \to Y$ from the real image to map domain, we implemented two canonical classification accuracy metrics: segmentation accuracy, and IoU accuracy, defined as follows:

$$\text{Segmentation Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP(\lfloor G_B(X_i) \rfloor)}{S}$$

where TP is the number of True Positive pixels in the quantized translated map $\lfloor G_B(X_i) \rfloor$ with respect to the ground truth quantized map $\lfloor Y_i \rfloor$, and $S = TP + TN + FP + FN =$ the total number

of pixels in each image. N is the size of the test set. Next,

$$\text{IoU Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{C}\sum_{j=1}^{C} IoU(\lfloor G_B(X_i)\rfloor_j)$$

where IoU is the class-wise intersection over union. Again, TP is evaluated with regards to $\lfloor Y_i \rfloor$. This time, the class index j determining pixel accuracy; a pixel p_1 in the class c_1 being positive is determined by the event $I_{c_1=j} == I_{c_1^*=j}$, where p_1^* and c_1^* are the corresponding ground truth pixel and class:

$$IoU(\lfloor G_B(X_i)\rfloor_j) = \frac{TP(\lfloor G_B(X_i)\rfloor_j)}{S_j}$$

Pixel classes c are considered after quantization.

Sensitivity to Noise (SN). This metric directly measures the model's sensitivity to adversarial noise. A mapping $G_A : Y \to X$ which is highly dependent on the specific embedded information in the segmentation images in the domain Y would yield a highly different reconstruction even with a slightly perturbed input; thus a low SN is indicative of an adversarially-robust model[2]:

$$SN(\sigma) = \frac{1}{N}\sum_{i=1}^{N}\|G_A(G_B(X_i) + \mathcal{N}(0,\sigma)) - G_A(G_B(X_i))\|_2$$

The SN metric is best computed as $\int_a^b SN(x)dx$; the authors use $a = 0$ and $b = .2$. However, this metric proves the most computationally costly, as formulaically, we are given point values of SN and therefore need to use Monte Carlo integration. The computation of each point value SN requires a new set of predictions on the test set, which is costly even with GPUs. We thus only report point values of SN evaluated at default values of σ.

Reconstruction Honesty (RH). Similar to SN, this metric is designed to assess the model's robustness to self-adversarial attack by examining reconstruction losses, with the intuition that a small modification to the intermediate image should not lead to wild fluctuations in the resulting reconstruction quality. As our implementation of this metric deviates substantially from the authors', this metric is discussed in more detail in section 3.3 below.

3.3 Proposed Metrics

We propose modifications to the authors' existing RH metric, as well as a novel version of our own. The authors specify

$$RH = \frac{1}{N}\sum_{i=1}^{N}\{\|G_A(\lfloor G_B(X_i)\rfloor) - Y_i\|_2 - \|G_A(G_B(X_i)) - Y_i\|_2\}$$

with the idea being that quantizing the intermediary $G_B(X_i)$ can class-semantically reflect any improperly embedded information. Yet, the definition appears domain-incoherent; we implement the modified metric

$$RH = \frac{1}{N}\sum_{i=1}^{N}\{\|G_A(\lfloor G_B(X_i)\rfloor) - X_i\|_2 - \|G_A(G_B(X_i)) - X_i\|_2\}$$

which is a difference between reconstruction losses as desired. We also propose (**P**RH) our own RH metric

$$PRH = \frac{1}{N}\sum_{i=1}^{N}\{\|G_A(Y_i) - X_i\|_2 - \|G_A(G_B(X_i)) - X_i\|_2\}$$

in which we calculate the difference between the translation quality of the real segmentation map Y_i and the reconstruction quality of the real image X_i. Whereas the authors propose a difference between reconstruction losses, our metric is more intuitive because we implicitly embed the translation quality objective that $G_B(X_i) \sim Y_i$. An honest mapping G_B is one which produces segmentation maps with high semantic similarity to its inputs, and minimal dependence on hidden high-frequency information. Tautologically, the ground truth Y_i is an honest representation of the desired segmentation map corresponding to X_i, so learning a $G_B(X_i) \sim Y_i$ is desirably honest.

[2]Note that there is a trade-off between SN and reconstruction quality, as G_A is a one-to-many mapping.

3.4 Wasserstein GAN

In Bashkirova et al.'s paper, the defense mechanisms are implemented on top of Least-Squares GANs (LSGANs) [8], which adopt the least-squares loss function for the discriminator. Compared to regular GANs, which use the sigmoid cross-entropy loss function, LSGANs are shown to be able to generate higher quality images and perform more stably during the learning process. However, there are still many other varieties of GANs. Wasserstein GANs (WGANs) [1] were introduced around the same time as LSGANs, and replace the discriminator model with a critic that scores the realness or fakeness of a given image (by approximating the Wasserstein distance). WGANs are less sensitive to model architecture and hyperparameter configurations, and the Wasserstein distance has the nice property that it is continuous and differentiable, providing useful gradients even after the critic is well-trained. To determine whether the defense mechanisms are sensitive to the type of GAN used, we also experiment with the defense mechanisms when implemented with WGANs.

3.5 Proposed Defense Mechanism: 4D Defense

We propose a new defense mechanism called the 4D defense. Specifically, we append a fourth square matrix (256 x 256) of zeros to each input image. This augmented image is then mapped by the generative network to another four dimensional image (4 x 256 x 256), of which the first three dimensions are intended to map to the semantics of the translated image and the fourth dimension is intended to hold any high-frequency, low-amplitude noise that may lead to adversarial attacks on the Cycle-GAN.

The intuition behind this defense mechanism is illustrated in Figure 1. The image left in Figure 1 illustrates the reconstructed and real images of a sample image following adaptive histogram equalization. Notably, the translation image has additional latent features (i.e. high frequency, low amplitude noise) which encodes the additional details of the real image in a format invisible to humans. Adversarial attacks occur when additive noise is added to the translated image, disrupting the image's latent features and reducing the image reconstruction accuracy.

The 4D defense mechanism attempts to encode the latent features in a new fourth channel of the image such that imperceptible noise added to the translation image only affects the three primary RGB channels, leaving the latent features undisturbed in the new fourth channel. The images on the right of Figure 1 illustrate the data stored in the fourth dimension of two sample images. In particular, the fourth dimension holds granular, low amplitude features which correspond to high detail areas on the original photos (such as home roofs).

In order to store the latent information in the fourth dimension, we would modify the cycle consistency loss equation 2 to penalize reconstruction errors in the RGB channels of the real image, but not the additional fourth dimension of the image. This intuitively makes sense because a semantic image reconstruction should only depend on the visible RGB channels and not the latent fourth channel. The modified cycle consistency loss equation is listed in 4.

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x))_i - x_i\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y))_i - y_i\|_1]. \tag{4}$$

In particular, the subscript i in equation 4 is the operator which restricts the resulting image to the first i channels. For the 4D defense, $i = 3$. We were unable to fully implement this new loss function, which should be conducted in future work. Our trained defense mechanism incorporates the appended matrix of zeros to construct the fourth channel, but uses the unmodified loss function listed in equation 3.

4 Results

4.1 Model Performance

We qualitatively assess the performance of the authors' four baseline models against ours. Figure 2 illustrates the four baseline models applied to one given input image alongside the image's translation, reconstruction, and noisy reconstruction. While the baseline CycleGAN provides the most comparable reconstruction to the real image (due to high dependency on the latent noise in the translated image), the CycleGAN models with noise provide the best noisy reconstruction (a

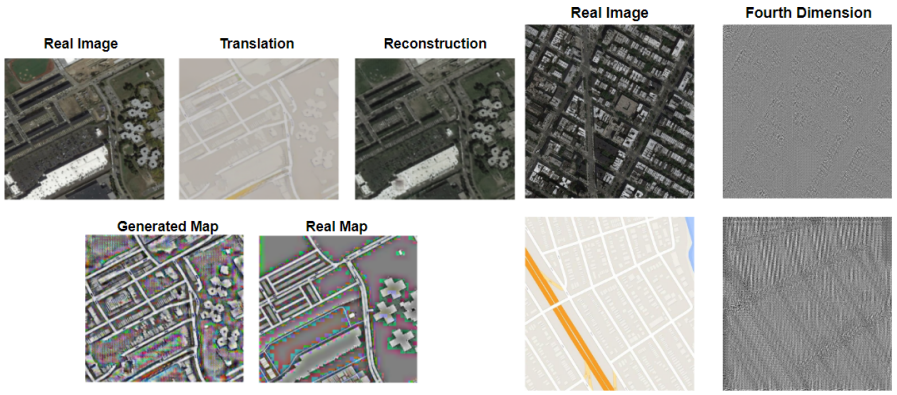

Figure 1: Left (taken from [3]): Top- A real image, the translation of the real image, and a reconstruction of the real image. Bottom- The translated image amplified by adaptive histogram equalization. The translation (left) and the ground truth translation (right). Right: Two sample real images (left) and the corresponding fourth dimension of the image translation displayed as a gray scale image(right);

more robust reconstruction at the cost of quality on undisturbed translation images). These results qualitatively mirror the authors' original results. Empirically, we also tested all the models mentioned in section 3.2.1; the results of the experiments are listed in Tables 1 and 2. Our results are consistent with [2] according to our metrics: of all models, the unmodified CycleGAN performs the worst according to the RH and SN metrics. While the CycleGAN achieves reasonable accuracy, such a balance of metrics confirms that the CycleGAN is vulnerable to self-adversarial attack: it is overfitting to achieve high accuracy by embedding hidden information in its translations, as revealed by the honesty metrics. It is visible from the table that nearly all CycleGAN + NG models achieve relatively high accuracy while maintaining reasonable translation honesty, with the exception of the WGAN.

Taking a closer look at our Proposed RH metric, we see that it is actually perhaps the best holistic indicator of an accurate, robust model. To see this, we focus on the three models trained with the 4D defense technique. These three models visibly performed much more poorly than all of the others, as can be seen from Figure 4. Yet the original RH and SN metrics indicate that they are relatively robust. On the other hand, they achieve the worst 3 scores according to our PRH metric (excluding the normal CycleGAN). This is expected, as the structure of our Proposed RH metric implicitly gauges translation quality. This is confirmed by the similarly poor accuracies achieved by the three 4D defense models. At the same time, the relatively good PRH values achieved by the CycleGAN + NG models confirms that PRH is also a good indicator of model honesty as intended.

4.2 Wasserstein GAN

We discovered that for WGANs, the noise defense technique proposed by the authors performs poorly based on our listed metrics. Furthermore, this conclusion is seen in the reconstructed pictures shown in Figure 3. Almost all reconstructed pictures in this setting filled streets with dark backgrounds and turned buildings and vehicles pink/purple. The poor performance of WGANs is further illustrated in Table 1 (entries colored red) in which the IoU and segmentation accuracy are very close to 0. Though we do not have an intuition for why this might happen, we can conclude that the defense metrics in the paper are indeed sensitive to GAN implementation.

4.3 Defense Mechanism: 4D Defense

The qualitative performance of the 4D defense mechanism is shown in Figure 4. The translated image has a color scheme which does not closely match that of the ground truth and also has distortions on the border of the image. Evidence for the poor image translation is shown by the low segmentation accuracy metrics of the 4D models in both Tables 1 and 2. Notably, Figure 4 illustrates that the reconstructed images have a similar color scheme to the original image, but have unnecessary motifs,

Figure 2: The models used the four baseline models from the original paper From left to right: the true input image (called A), the ground truth of the translation of A, the actual translation of A (called B), the reconstruction of A from B, and the reconstruction of A from B with Gaussian noise. [2]

Figure 3: As shown here, the noise defense techniques proposed by the authors lead to poor reconstructions on the WGANs.

particularly around the image borders. Evidence for the poor reconstruction performance is shown by the high Proposed RH metrics in Tables 1 and 2, highlighted in red.

5 Conclusion

We reproduced the adversarial defense algorithms for CycleGANs proposed in [2]. Additionally, we produced qualitative images and qualitative metrics which supported the authors' claim that the defense mechanisms make the CycleGAN models more robust to adversarial attacks. We conducted hyperparameter searches to determine the robustness of the authors' proposed algorithms. We note that the authors' noise and guess defense mechanisms consistently outperform the CycleGAN baseline across our metrics, indicating the defense mechanisms generally defend against adversarial attacks. Notably, we identified that the noise defense is not effective against WGAN loss functions, which suggests an area for future research. Furthermore, we improved upon the authors' work by modifying their proposed RH metric. Our redeveloped RH metric provides more accurate measurements of model performance, supported by qualitative image analysis. Finally, we propose a novel defense mechanism, 4D defense, to increase the robustness of image semantics to the effects of imperceptible noise. While the 4D defense did not perform well relative to the authors' proposed defenses, our implementation of the 4D defense would be improved by modifying the cyclic consistency cost function, which is an area of future work.

References

[1] M. Arjovsky and L. Bottou, “Towards Principled Methods for Training Generative Adversarial Networks,” in International Conference on Learning Representations, 2017.

[2] D. Bashkirova, B. Usman, and K. Saenko, “Adversarial Self-Defense for Cycle-Consistent GANs,” in Neural Information Processing Systems, 2019.

[3] C. Chu, A. Zhmoginov, and M. Sandler, “CycleGAN, a Master of Steganography,” in Neural Information Processing Systems, 2017.

[4] H. Dong et al., “Soft-Gated Warping-GAN for Pose-Guided Person Image Synthesis," in Neural Information Processing Systems, 2018.

[5] I. Goodfellow et al., “Generative Adversarial Nets,” in Neural Information Processing Systems, 2014.

[6] L. Karacan, Z. Akata, A. Erdem and E. Erdem, “Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts," in European Conference on Computer Vision (ECCV), 2018.

[7] M. Liu, T. Breuel, and J. Kautz, “Unsupervised Image-to-Image Translation Networks,” in IEEE International Conference on Computer Vision (ICCV), 2017.

[8] X. Mao et al., “Least Squares Generative Adversarial Networks,” in IEEE International Conference on Computer Vision (ICCV), 2017.

[9] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” arXiv preprint arXiv:1411.1784, 2014.

[10] S. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for Data: Ground Truth from Computer Games,” in European Conference on Computer Vision (ECCV), 2016.

[11] X. Sun, H. Xu, and K. Saenko, “A Two-Stream Variational Adversarial Network for Video Generation,” arXiv preprint arXiv:1812.01037, 2018.

[12] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka, “ArtGAN: Artwork Synthesis with Conditional Categorical GANs,” in IEEE Conference on Image Processing (ICIP), 2017.

[13] T. Wang et al., “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," in IEEE Conference on Computer Vision (ICCV), 2018.

[14] J. Zhu, T. Park, P. Isola, and A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” in IEEE International Conference on Computer Vision (ICCV), 2017.

Appendix

5.1 Original Metrics

The default CycleGAN model uses the following architecture and parameters:

- General architecture – ResNet with 9 residual block layers
- Discriminator architecture – 3-layer PatchGAN with patch size 70 x 70
- Weight initialization – Gaussian
- GAN objective – LSGAN
- Optimizer – Adam with momentum 0.5
- Learning rate – 0.0002 with linear policy
- Trained for 200 epochs

5.2 4D Defense: Supplementary Figures

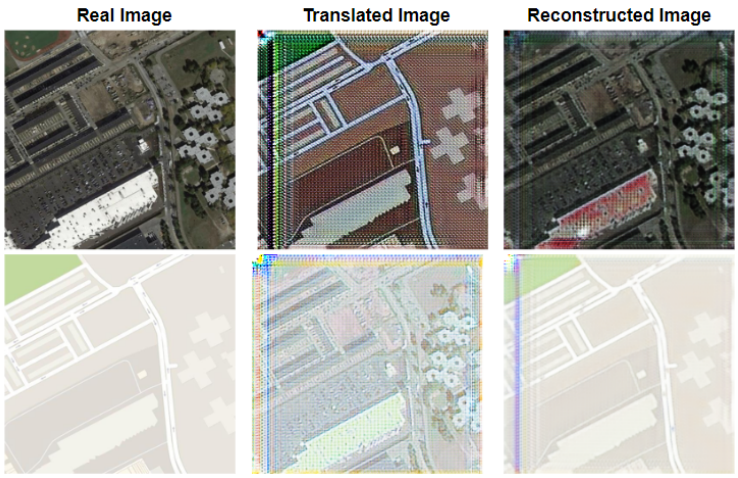

Figure 4: Sample image after translation and reconstruction with the proposed defense mechanism

5.3 Additional Results

In tables 1 and 2, note that NG means $Noise + Guess$. The baseline CycleGAN + NG model uses hyperparameters $\sigma = 0.06, L_p = linear, \lambda_A = 1, \lambda_B = 2, \lambda_{guess} = 1, LSGAN, M = 0.5, L_r = 0.0002$.

Figure 5: Using different std values for noisy + guess defenses with CycleGANs.

Table 1: Metrics (RGB Images)

Model	RH	Proposed RH	Seg. Acc.	IoU Acc.	SN(sigma)
CycleGAN	25.611	53.740	0.529	0.065	64.875
CycleGAN + Noise	2.522	38.463	0.555	0.068	40.642
CycleGAN + Guess	7.827	34.176	0.471	0.058	64.874
CycleGAN + NG ($L_p = step$)	1.458	26.100	0.486	0.060	31.382
CycleGAN + NG ($L_p = cosine$)	1.485	28.041	0.474	0.058	34.386
CycleGAN + NG ($\sigma = 0.04$)	1.770	20.005	0.523	0.065	46.420
CycleGAN + NG ($\sigma = 0.10$)	0.230	16.983	0.478	0.029	46.788
CycleGAN + NG ($M = 0.6$)	-1.569	17.497	0.469	0.058	49.001
CycleGAN + NG ($\lambda_A = 1, \lambda_B = 1$)	-0.470	19.560	0.437	0.056	48.959
CycleGAN + NG ($\lambda_A = 2, \lambda_B = 1$)	2.784	38.195	0.487	0.061	47.571
CycleGAN + NG ($\lambda_A = 1, \lambda_B = 2, WGAN$)	0.015	-3.690	0.006	0.0001	15.311
CycleGAN + Guess (4D defense)	2.405	49.456	0.220	0.027	24.654
CycleGAN + Noise (4D defense)	0.101	91.288	0.023	0.003	5.602
CycleGAN + NG (4D defense)	0.218	105.000	0.147	0.018	10.970

Table 2: Metrics (Grayscale Images)

Model	RH	Proposed RH	Seg. Acc.	IoU Acc.	SN(sigma)
CycleGAN	14.714	31.218	0.299	0.007	37.262
CycleGAN + Noise	1.498	22.222	0.379	0.009	23.233
CycleGAN + Guess	4.529	19.856	0.219	0.005	37.267
CycleGAN + NG ($L_p = step$)	0.855	15.283	0.282	0.006	18.009
CycleGAN + NG ($L_p = cosine$)	0.844	16.262	0.275	0.006	19.662
CycleGAN + NG ($\sigma = 0.04$)	0.672	11.523	0.351	0.008	26.350
CycleGAN + NG ($\sigma = 0.10$)	-0.932	10.114	0.294	0.007	28.282
CycleGAN + NG ($\lambda_A = 1, \lambda_B = 1$)	-0.321	11.307	0.233	0.006	27.992
CycleGAN + NG ($\lambda_A = 2, \lambda_B = 1$)	-1.623	-22.133	0.273	0.006	27.206
CycleGAN + NG ($\lambda_A = 1, \lambda_B = 2, WGAN$)	-0.008	2.105	0.008	0.0001	8.786
CycleGAN + Guess (4D defense)	-1.375	-28.504	0.052	0.001	14.168
CycleGAN + Noise (4D defense)	-0.060	-52.824	0.003	0.00007	3.215
CycleGAN + NG (4D defense)	-0.128	-58.799	0.015	0.0003	6.274

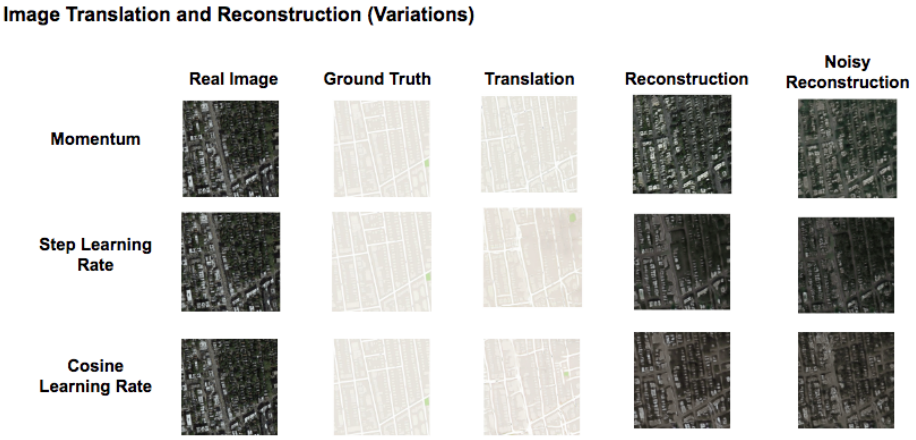

Figure 6: Using different learning rates with noisy + guess defenses with CycleGANs.

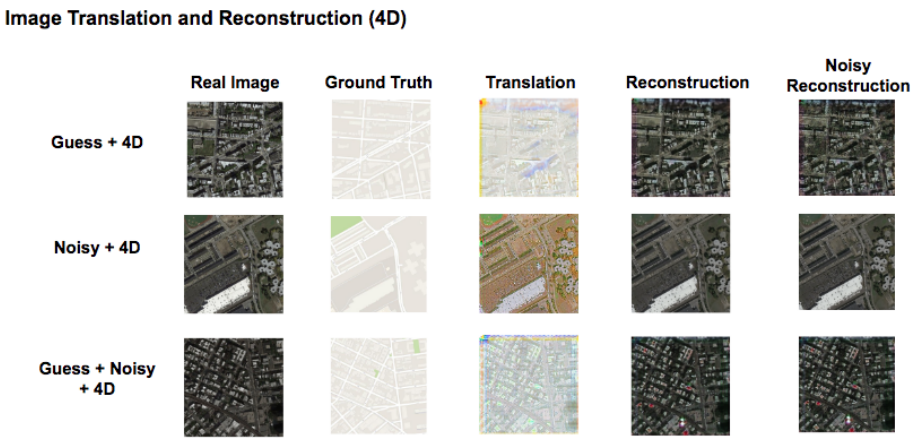

Figure 7: Here are some image results for the proposed 4D defense stacked on top of guess/noisy defenses.