Jekyll: Attacking Medical Image Diagnostics using Deep Generative Models

Neal Mangaokar[1], Jiameng Pu[1], Parantapa Bhattacharya[2], Chandan K. Reddy[1], Bimal Viswanath[1]

[1]Virginia Tech   [2]University of Virginia

{neal1998, jmpu, ckreddy, vbimal}@vt.edu, parantapa@virginia.edu

Abstract—Advances in deep neural networks (DNNs) have shown tremendous promise in the medical domain. However, the deep learning tools that are helping the domain, can also be used against it. Given the prevalence of fraud in the healthcare domain, it is important to consider the adversarial use of DNNs in manipulating sensitive data that is crucial to patient healthcare. In this work, we present the design and implementation of a DNN-based image translation attack on biomedical imagery. More specifically, we propose *Jekyll*, a neural style transfer framework that takes as input a biomedical image of a patient and translates it to a new image that indicates an attacker-chosen disease condition. The potential for fraudulent claims based on such generated 'fake' medical images is significant, and we demonstrate successful attacks on both X-rays and retinal fundus image modalities. We show that these attacks manage to mislead both medical professionals and algorithmic detection schemes. Lastly, we also investigate defensive measures based on machine learning to detect images generated by *Jekyll*.

1. Introduction

As we make rapid advances in deep learning/AI, it is important to understand the associated security implications. Attackers can no longer be assumed to have limited algorithmic intelligence [1]. An emerging threat is that of "deepfakes", or AI generated synthetic content that appears convincingly real [2]. Deepfakes are enabled by deep generative models such as Generative Adversarial Neural Networks (GANs) [3]. Generative models can produce photorealistic fake images [4], [5], and convincing fake videos [6]. Recently, the threat of deepfakes has been largely discussed in the context of the Web, where they can be used to create fake accounts, fake pornographic images of celebrities, images of people doing things they never did to spread misinformation, and manipulate elections [7]. In this work, we investigate threats posed by deepfakes in the healthcare domain, *i.e.* how bad actors can make use of generative schemes to attack critical workflows in our healthcare framework.

Healthcare spending is huge in many developed countries, and the system is already fraught with fraud [8]–[13]. Prior work has highlighted exploitable vulnerabilities in the healthcare domain. The industry has employed poor security practices in securing sensitive patient data such as biomedical images [14]–[16]. In 2019, Mirsky et al. demonstrated attacks that compromise biomedical data management pipelines [16]. As our healthcare system is susceptible to bad actors, it is important to investigate new threat vectors driven by technological advances.

We propose an attack framework called *Jekyll* that leverages generative deep learning to derail one of the most important decision processes in the medical domain—*medical diagnosis based on biomedical image analysis*. *Jekyll* is based on a Generative Adversarial Neural Network (GAN). Our key insight is to use an image-to-image style transfer approach for our attack. *Jekyll* takes as input a biomedical image of a victim, and translates it to produce a new (fake) image that changes the "style" by injecting an attacker-chosen disease into the image, while preserving the "content" or identity of the victim. The targeted condition does not reflect the real health condition of the patient. Therefore, the key outcome is to produce a "deepfake" image that can cause both a human (medical professional), and an algorithm to misdiagnose the health condition [17]–[19]. This is because the produced image shows human-perceptible signs of the targeted condition, which is also sufficient to mislead automated algorithmic schemes.

An incorrect diagnosis can lead to potential life-threatening situations for the patient, unnecessary healthcare costs, and wasted healthcare resources. The attacker may be motivated by financial gain. For example, a malicious clinic can trigger misdiagnosis and force an insurance provider to pay for unnecessary procedures. The attacker may also be motivated by criminal intent to cause harm to an individual (or group of individuals), or to disrupt and damage a particular healthcare framework [16], [20].

A key aspect of *Jekyll* is the capability to perform *controlled* creation of the fake image. Apart from being able to inject an attacker chosen disease condition, it can be done while preserving the "identity" of the patient. This is important because biomedical images are known to contain patterns that are unique to their patients. Preserving identity makes the attack much harder to detect because the image will appear to belong to the victim and will not seem abnormal. To make the attack more damaging, our tool can also be used during repeat visits by the victim to inject disease patterns that mimic the natural progression of the disease.

Attacks using *Jekyll* are realized by training on publicly available medical image datasets. Our attack only requires datasets annotated with health conditions and (anonymized) patient IDs. More importantly, *Jekyll* only requires a single image of the victim to generate a misleading *fake* image.

Key contributions of our work include the following: (1) We design and implement a GAN-based tool called *Jekyll* that can inject an attacker chosen disease condition into a victim's image, while preserving their identity.

139

(2) We demonstrate the feasibility of the attack on two popular biomedical image modalities—X-rays and retinal fundus images. Using publicly available medical datasets, *Jekyll* is used to inject Cardiomegaly, and Pleural Effusion health conditions into chest X-rays of healthy patients. For retinal images, we demonstrate injection of the Diabetic Retinopathy condition.

(3) We show that attacks powered by *Jekyll* can be sustained over time. As patients make repeat visits to a hospital, *Jekyll* can be used to inject disease conditions that match the expected progression of a disease.

(4) The effectiveness of attacks by *Jekyll* is evaluated by both (a) machine learning algorithms and image quality metrics, and (b) medical professionals. Our user study shows that medical professionals are convinced of the presence of targeted disease conditions in the fake images, and that they are unable to distinguish between real and fake images.

(5) Finally, we explore defensive schemes. We investigate two machine learning-based detection schemes: (1) *Blind detection*: assumes no access to fake images, and no knowledge of attacker's model for training, and (2) *Supervised detection*: assumes access to both real and fake images for training. We show that supervised detection schemes are highly effective, but also vulnerable to evasion schemes that modify *Jekyll* to bypass detection.

2. Background and Related Work

Problem motivation. Healthcare spending is huge in many developed countries. In 2017, the US spent 17% of its GDP on healthcare [21]. Not surprisingly, given the money involved, the healthcare system is already fraught with fraud [8]–[13]. Different entities/actors in the system—medical institutions (large hospitals as well as small clinics), medical practitioners (e.g., physicians, radiologists), health insurers, all have an incentive to engage in fraud and benefit financially [8]. For example, a clinic can bill patients for unnecessary procedures or medication [22]. On one hand, hospitals are known to inflate the cost of medical care to overcharge patients, while on the other, insurers have an incentive to reduce payout [23], [24]. The attacker may also be motivated by a criminal intent to cause harm to an individual (or group of individuals), or to disrupt and damage a particular healthcare framework [20]. Overall, security practices implemented in healthcare systems are lacking, making them vulnerable to attacks that compromise the availability and integrity of medical data [14], [15]. These trends motivate us to explore potential threats by malicious actors that leverage technological advances in machine learning to engage in hard-to-detect medical fraud.

Biomedical images. Medical imaging is a crucial component of any health care framework. Biomedical images taken using specialized instruments capture interior anatomical structures of a human body. By analyzing these images, medical practitioners can monitor diseases and prepare treatment plans, often without requiring any invasive procedures [25]. To demonstrate our attack, we focus on two widely used image modalities, X-rays, and retinal fundus images. X-ray images are widely used to assess a range of injuries (e.g., damaged bones), and health conditions such as heart disease, breast cancer, and collapsed lungs. Fundus photography helps to capture an image of the back of the eye. Fundus photographs of the retina [26] are used by ophthalmologists to detect Diabetic Retinopathy, a condition that could lead to vision loss in patients with diabetic mellitus [26].

Attacks proposed in this work, while demonstrated on X-ray and retinal fundus imagery, are theoretically applicable to other 2D image modalities, *e.g.*, Ultrasound, and 2D MRI scans [25]. In Section 3.1, we also explain potential application to 3D modalities. Also, unlike other images, biomedical images are usually highly standardized in terms of anatomical position and exposure [27]. This makes it easier to learn patterns in anatomical structures and makes them more vulnerable to attacks discussed in this work.

Adoption of ML for healthcare decisions. Given the availability of large medical image datasets, and advances in deep learning, it is expected that algorithms will play a significant role in aiding healthcare decisions [26]. Deep learning schemes that analyze medical images can help doctors spot health conditions that may be otherwise hard to identify even by a trained professional [17]. Insurance providers may also leverage algorithms to automate verification of diagnoses, before making reimbursements [8]. Recently, the community has seen rapid advances in algorithmic decision making for medical imaging tasks that even surpass human performance. Recently, the U.S. Food and Drug Administration approved an AI algorithm to screen chest X-rays for collapsed lung (or Pneumothorax) [28]. In fact, AI-based systems are already being tested/deployed to assess diabetic blindness [29], detect chromosomal abnormalities [30], and pancreatic cancer [31]. Recent examples also include ML schemes that perform well on breast cancer detection [32], skin cancer classification [33], arrhythmia detection [34], hemorrhage identification [35], and diabetic retinopathy detection [36].

Hence, a successful attack that aims to mislead any medical decision process, should also fool an ML scheme designed for the same decision process. Otherwise, even if it can mislead a medical professional, it could be thwarted by an ML scheme.

***Jekyll* vs attacks using adversarial samples.** Given an input image, one can add carefully crafted adversarial perturbations that are imperceptible to humans, such that the perturbed input triggers misclassification when fed to a model [37]. Therefore, one approach to mislead ML-based diagnostics tools is to craft adversarial samples of biomedical images—given a chest X-ray of a healthy patient, craft an adversarial input that fools the ML model to predict a disease condition [8]. However, today, medical images still undergo visual examination by professionals—radiologists can identify that an adversarial X-ray image (targeting a disease condition) still looks healthy, thus rendering the attack ineffective. *Hence, we propose attacks that produce images containing visually perceptible changes that can mislead both a human (more importantly, a medical professional) and an ML model.*

Prior work on misuse of ML/AI. Most prior work at the intersection of ML and security primarily focused on attacks against ML systems [38]. But, recently, there is emerging interest in understanding new attacks enabled by ML/AI [1]. In the non-medical space, research has demon-

strated AI techniques to break CAPTCHA systems [39], generate convincing fake reviews [40]–[42] and email content [43], control voice assistants (e.g., Google Assistant, Alexa) [44], [45], extract private information from collaborative learning systems [46], attack anonymity systems (Tor) [47], [48], and to automate DoS attacks via trace synthesis [49].

In the medical domain, there is limited work on understanding threats posed by AI. Kohli et al. proposed an Iris presentation attack using a DCGAN [50]. Their attack specifically focused on Iris biometric systems and is not generally applicable. In more closely related concurrent work from 2019, Mirsky et al. proposed CT-GAN [16], a framework that uses deep learning to tamper with 3D medical imagery (CT images) to add or remove signs of medical conditions. Compared to both these works, we propose a generic attack that is applicable to multiple diseases, multiple modalities (X-ray, retinal fundus), and requires significantly less effort from the attacker to successfully mislead diagnostic processes. We discuss CT-GAN in more detail in Section 3.1.

3. Attacking Medical Diagnostics

3.1. Threat Model and Overview of Approach

Threat Model. The victim, a patient, visits a medical imaging lab, and obtains a biomedical image of some type, *e.g.,* X-ray, retinal Fundus. Unknown to the victim, the attacker obtains access to the victim's image, and "translates" it to a version with an attacker-chosen disease condition (that does not reflect the actual health situation of the victim). After translation, the original image is removed from the system by the attacker and is not seen by anyone else. For example, given an X-ray of a healthy patient, the attacker generates a new image indicating an abnormal heart condition when examined by either a medical professional or a machine learning algorithm. Figure 1 illustrates the attack scenario. It is important to note that medical professionals will analyze the image for presence of any diseases. As the generated image is examined by a professional, traditional adversarial sample attacks, or naïve attacks that tamper with the diagnostic end-results are ineffective. We make the following assumptions:

First, the attacker can access and alter victim's medical images by compromising medical information systems. Medical images are typically managed through a Picture Archiving and Communication System (PACS) [51]. The attacker can either access the data-at-rest, *e.g.,* when it is stored on a PACS server, or when data-is-in-motion, *e.g.,* by intercepting network traffic or by reading from volatile memory. Real world deployments of PACS are known to follow poor security practices, *e.g.,* misconfigurations, using default credentials, easy access by insiders, rare security patch updates, and lack of encryption support [14], [15]. These flaws cause the PACS systems to be highly vulnerable to social engineering and physical insider attacks, thereby violating data integrity. In fact, Mirsky et al. [16] demonstrated a successful pen-test using a Raspberry Pi that was able to compromise a hospital's PACS framework, making it feasible to intercept scans and credentials of staff members in plaintext, and install a malware that allowed for a *man-in-the-middle* attack.
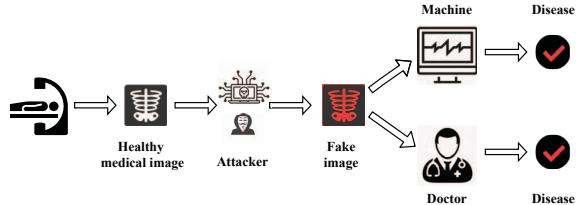

Figure 1. Overview of attack scenario.

PACS servers are also exposed to and thus accessible via the internet [51]. This exposure amplifies the susceptibility to attacks via web portals [14].

Second, the attacker can leverage publicly available medical image datasets to build ML models for the attack. There are large datasets covering different image modalities, e.g., X-ray [52], retinal fundus images [26], MRI [53], CT [54], [55], and PET [53]. The attack requires annotations that capture anonymized patient IDs, health conditions, *i.e.* whether patient is healthy or has some disease(s), and any available characteristics such as the stage or severity of the condition. *Note that the attack does not require any disease segmentation masks for these images, e.g., a marked portion of the image that shows an area affected by disease.*

Third, the attacker has no access to any previous medical images of the victims. The attack only requires a single image from the victim, which can be accessed after the patient undergoes a medical imaging procedure.

Lastly, no knowledge is assumed regarding how the image is analyzed by the entity responsible for the victim's healthcare. Images could be analyzed by a medical professional or a machine learning algorithm (or both). Also, we do not assume any query access to the machine learning algorithm evaluating the victim's health condition. Such an access is not required in our attack because the attack visibly injects the targeted disease in the image, unlike traditional adversarial inputs where imperceptible adversarial noise is added to fool classifiers [56].

Attack goals. Three main goals include the following:
1. Translate a biomedical image to a new one that indicates an attacker chosen health condition. The targeted condition does not reflect the real health condition of the patient. In the rest of the paper, we use the phrase "disease injection" to refer to this image translation process, and the generated image is called the *fake image*. While we primarily focus on injecting a disease condition, our methods can also be used to translate an image with a disease condition to one that appears healthy.
2. Inject disease while preserving identity of the victim. For a successful attack, one must not only inject the disease, but must also ensure that the fake image reflects the "identity" of the victim. Biomedical images typically contain patterns that are unique to their owners, and such personal signatures can be used to verify the identity of the owners [57]. This applies to modalities such as X-rays, and retinal images that carry unique signatures of the patients. If the attacker used a diseased X-ray from another patient or generated a chest X-ray image that fails to preserve the anatomical characteristics of the victim, the image could be flagged by an identity verification algorithm, or by a doctor on visual examination. In such cases, the doctor or algorithm can compare the generated

image against previously submitted images of the patient to verify the identity, thus rendering the attack ineffective. *3. Sustaining the attack over time.* Patients typically undergo repeated examinations to follow up on a health condition. This provides an opportunity for the attacker to continue to manipulate the system. To enable subsequent attacks, the attacker would need to control the disease injection process to reflect the natural progression of a health condition. Otherwise, there is a risk of raising suspicion and being caught. Such repeated attacks can be devastating for the patient and lead to wasted resources in the health-care system. For this goal, we propose methods to sustain the attack over time.

Comparison to CT-GAN [16]. The only closely related work is CT-GAN, a deep-learning framework for 3D medical image tampering. The key differences between *Jekyll* and CT-GAN are as follows: *(1) Practicality and efficiency.* To inject a chosen disease condition, CT-GAN requires at least 14 complex steps, including extensive pre-processing steps, identifying the region for disease injection, and multiple manual touch-ups to make the tampered region look realistic. *Jekyll* is more efficient to use, as it only requires the single translation step of passing the image through the generator. *Jekyll* is also more practical for a generalized bad actor, as it requires no knowledge of medically viable regions for disease injection. *(2) Generalization to non-localized conditions.* CT-GAN uses an "in-painting" scheme to inject a disease into a specific image region. However, this not only requires that the chosen region be a medically viable location for the disease, but also requires the disease to be localized in the said region. *Jekyll*, on the other hand, learns to inject the disease as a whole into the image. This enables injection of diseases that are spread unevenly over multiple anatomical regions *e.g.,* injecting diabetic retinopathy into retinal fundus images. It is unclear if CT-GAN can work with such modalities. *(3) Image Dimensionality.* CT-GAN is intended for use with 3D medical image modalities *e.g.,* CT scans. Disease injection is performed by extracting a 2D image slice from the middle of 3D DICOM imagery for the CT scan, applying the in-painting technique, and re-inserting the modified slice. *Jekyll* performance is demonstrated on 2D medical imagery but can potentially be extended to 3D modalities by performing single-step translation on a similarly extracted middle slice. Overall, *Jekyll* and CT-GAN demonstrate new threats facing our healthcare system.

3.2. Attack Methodology

We use a Generative Adversarial Network (GAN) [3] for the attack. A GAN has two primary components, a *generator* and a *discriminator* that are trained in an adversarial process. Given a dataset of images, the generator learns to generate synthetic images that mimic the distribution of the dataset. The discriminator learns to decide whether an image produced by the generator looks real (i.e., belongs to the true data distribution) or fake. The two components are trained using a minimax objective where the generator aims to produce fake images that are indistinguishable from real images, while the discriminator aims to rightly distinguish between real and fake images. GAN variants have shown impressive results for high quality image generation tasks [4], [5], [58].

Prior work using GANs in the medical domain mainly focused on non-adversarial scenarios, *e.g.,* for data augmentation [59]–[62], de-identification [50], anomaly detection in data [63], feature extraction [64], [65], and image segmentation [66]–[71]. Our goal is not to propose a new GAN model to advance the state-of-the-art in biomedical image generation. Instead, we show how a GAN-based approach can be used to launch attacks against medical diagnostics.

While many GANs have been proposed, not all of them are suitable for the attack. In a vanilla GAN, the generator takes as input a noise vector z drawn from some distribution $p_z(z)$ (e.g., normal distribution) to generate an image, while the discriminator provides feedback to improve the generation process. Different input vectors will produce different images, but in general it is hard to reverse engineer how the noise space maps to specific semantic properties of generated images, e.g., to represent a disease or identity of a patient. Therefore, this does not fit our scenario—having *only* a latent vector as input makes it hard to *control* the generation process. Another challenge is that the attacker only has a single image of the victim. This rules out approaches that train a GAN on past images of the victim to generate identity preserving images. Instead, we propose to learn from medical images of other patients (publicly available data) for the attack.

Jekyll: **Our attack framework.** In this section, we present *Jekyll*, a GAN-based image style transfer model for attacking medical diagnostics. A style transfer GAN that takes an input image as a *condition*, and "translates" it to a version that preserves the *content*, while changing the *style* [72]. In our context, content includes image characteristics that capture identity of the patient, while style captures the health condition of the victim. Note that using an image as input provides more control over the generation process, compared to a vanilla GAN.

One challenge is that an image-to-image translation GAN requires paired input-output data. For example, Pix2Pix GAN [72] learns from paired data to transfer style. In our setting, this would require a pair of images belonging to each patient, i.e., one with no disease, and the other with a disease. Such paired data is usually not publicly available and can be challenging for the attacker to obtain. Instead, we propose to do style transfer from *unpaired collections of images*, *e.g.,* a set of images of arbitrary patients with no disease, and another set with a disease (again an arbitrary set of patients). The recently proposed CycleGAN [73] best fits this scenario, and we propose to build on top of this approach. *A key advantage of our approach is that we will automatically learn characteristics of the disease (style), and identity (content) from the image collections, without requiring any human intervention or image segmentation masks (that highlight regions indicative of disease or those that capture identity of a patient). This enables a single-step disease injection attack, unlike prior work (CT-GAN).*

Jekyll **design.** Let X be the domain of images having health condition C_X, and Y be the domain of images diagnosed with another condition C_Y. Training samples in X are $\{x_i\}_{i=1}^N$, and in Y are $\{y_i\}_{i=1}^M$. Our goal is to translate images from X to Y, *i.e.* inject a new disease condition while preserving the identity of the patient.

Without loss of generality we consider the victim to be a healthy person. We consider health condition C_X to denote a healthy condition, i.e., with no disease, and C_Y to denote a *single* real disease. *Jekyll* can also be used for disease removal, in which case C_X would include patients with a disease, and C_Y will be healthy patients.

An optimal image translator can translate images in X to ones that match the distribution of images in Y. But because of the unpaired data setting, there are infinite possible mappings from X to Y, and we want to produce a specific mapping—one that preserves the identity, while injecting a disease. We will later explain that this will require the use of two generators, and two discriminators to enable transfer from one domain to the other. One generator, $G : X \rightarrow Y$, transfers images from X to Y, and another generator $F : Y \rightarrow X$ transfers images from Y to X. One discriminator D_Y tries to distinguish real images in Y from images generated by G, another discriminator D_X tries to distinguish real images in X from images generated by F. Below, we explain how *Jekyll* enables style transfer for the attack. *Jekyll* is trained to optimize an objective function that includes the following loss terms.

Adversarial process. This part captures the basic *adversarial loss* for the GAN. A generator produces images that fall in a certain domain, and the discriminator tries to differentiate between generated and real images. We use a least-squares GAN loss [73] for our adversarial loss to stabilize training and to improve image quality. It is computed as:

$$L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data(y)}}[D_Y(y)^2]$$
$$+ \mathbb{E}_{x \sim p_{data(x)}}[(D_Y(G(x)) - 1)^2] \quad (1)$$

We use a similar adversarial loss function for the reverse direction, $F : Y \rightarrow X$ using the discriminator D_X. The final adversarial loss is as follows: $L_{adv} = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, X, Y)$. The two generators try to minimize this objective, while the two discriminators try to maximize it.

Disease Injection. In theory, the adversarial loss should be sufficient to inject a disease. However, we empirically observe that this is not the case. Disease characteristics in biomedical images can be subtle, e.g., small change in heart shape in a Chest X-ray, minor changes in a retinal vascular pattern. It is simple for the GAN to trivialize the style differences and simply replicate the input image. Thus, we incorporate an additional *disease loss* term to enforce disease injection.

A pre-trained disease classifier S is used to calculate the disease loss and provide additional feedback to the generator. Generated images from G are fed into S to obtain a prediction probability $S(G(x)|C_Y)$ for belonging to disease condition C_Y. If $G(x)$ receives a high prediction probability to be in class C_Y, we add a small penalty, and a high penalty if the prediction probability is low. This pushes *Jekyll* to correctly inject the targeted disease condition. We define

$$L_{disease} = \mathbb{E}_{x \sim p_{data(x)}}[l(S(G(x)|C_Y), C_Y)] \quad (2)$$

where l is a cross entropy function.

Preserving identity. As these are unpaired images, the adversarial loss term can map one image in X to any random point in domain Y. This is not desirable, because we want to find a mapping that preserves identity. To reduce the space of possible mappings in Y, we draw on work from CycleGAN and apply a *cycle consistency loss* to the GAN. Put simply, an image x when translated to Y and reconstructed back to domain X should be mostly similar to the original x, i.e., $F(G(x)) \approx x$. Similarly, there is a reverse cycle loss for translations from Y to X as well. The cycle loss is computed as:

$$L_{cycle} = \mathbb{E}_{x \sim p_{data(x)}}[\|F(G(x)) - x\|_1]$$
$$+ \mathbb{E}_{y \sim p_{data(y)}}[\|G(F(y)) - y\|_1] \quad (3)$$

While we expect the cycle loss to find mappings to the other domain that are easier to reconstruct, it still lacks a concrete notion of identity. This is because the cycle loss formulation does not explicitly characterize what defines the identity of the patient. We argue that cycle loss is not sufficient to preserve identity all the time. To better preserve identity, we propose an additional *identity loss* term defined as perceptual loss given by

$$L_{identity} = \mathbb{E}_{x \in p_{data}(x)}[\|E(x) - E(G(x))\|_1$$
$$+ \|E(F(G(x))) - E(G(x))\|_1)] \quad (4)$$

where $E(.)$ represents features extracted from a specific layer in a pre-trained identity classifier. Recall that attacker has a single image of the victim, so it is hard to train an identity classifier that includes all victims. *Using a perceptual loss as opposed to a classification loss (as in the disease loss term) helps to overcome this issue. Perceptual loss allows us to use an identity classifier trained on any available set of patients because we only use features from an internal layer.*

Finally, the overall loss is computed as:

$$L(G, F, D_X, D_Y) = \lambda_{adv}L_{adv} + \lambda_{disease}L_{disease}$$
$$+ \lambda_{identity}L_{identity} + \lambda_{cycle}L_{cycle} \quad (5)$$

and the associated weight terms control the extent to which each property is enforced.

Sustaining the attack over time. Ideally, when victims undergo repeated examinations, disease injections should match the expected progression of a disease, *e.g.,* disease becoming severe over time. We present two ways in which *Jekyll* can be used to enable such repeated attacks:

(1) Attacker can use publicly available datasets that capture different stages of the disease in question, and create multiple *Jekyll* models, each one trained to inject a specific stage of the disease into the patient's image. In fact, such datasets exist—we use a dataset of retinal fundus images to inject different stages of Diabetic Retinopathy (Section 5.1.4).

(2) If there is no data capturing progression of a disease, then we propose a simple alternative solution. Given a dataset capturing a certain (late) stage of a disease, and a healthy stage, attacker can inject intermediate stages of the disease using simple *linear interpolation* over the available images. More specifically, the attacker will train a single *Jekyll* model to translate a non-disease image to a disease stage (for which data is available). Next, given a non-disease image I_{nd} belonging to a victim, and a disease injected image, I_d produced by *Jekyll*, attacker can use linear interpolation to approximate intermediate stages

(represented by I_f) of the targeted disease as follows: $I_f = \alpha \cdot I_d + (1 - \alpha) \cdot I_{nd}$. Here α represents the degree of disease injection. Such injection is possible because *Jekyll* produces output images that are perfectly aligned with input images. In Section 5.1.4, we show how one can produce convincing attack images capturing intermediate stages of Cardiomegaly (heart condition) using this approach. However, we acknowledge that such interpolation schemes may not be meaningful for all disease conditions.

***Jekyll* model architecture.** *Generator and Discriminator.* Architecture for both generator and discriminator is inspired by CycleGAN [73], and we build on top of a publicly available implementation from GitHub [74]. Input and output image resolutions of the generator are 256×256. Discriminator is based on a 70×70 Patch-GAN [72], that decides whether 70×70 overlapping image patches are real or fake, resulting in a $32 \times 32 \times 1$ dimensional output. This has been shown to outperform a discriminator that evaluates the entire image to determine whether it is real or fake. More details of the generator and discriminator architecture are in Table 12 in Appendix C. Note that *Jekyll* can be adapted to produce higher resolution images by borrowing architectural elements from PG-GAN [4]. PGGAN can produce high resolution images by starting from a low resolution version, and progressively increasing the size (layers) of the network.

Disease and Identity Classifiers. Recall that *Jekyll* requires pre-trained disease and identity classifiers to preserve identity while injecting disease. We use the same architecture for both classifiers, but they are trained differently depending on the dataset.[1] Disease classifier is a binary classifier predicting condition as non-disease and the targeted disease. Identity classifier is a multi-class classifier predicting the identity of a person. Both classifiers use the DenseNet-121 model architecture [75]. For both models, we replace the last classification layer of DenseNet-121 with a dense layer of 256 neurons, a dropout layer with rate of 0.5, followed by a final classification layer (that fits our task). To compute the identity loss, we extract the output of the convolution layer before the last dense block in DenseNet-121. More details of *Jekyll*'s architecture are in Appendix C.

Alternative architectures for *Jekyll*. Our techniques behind *Jekyll* (to inject disease and preserve identity) can be applied to other image-to-image translation GANs as well. In Section 5.1.5, we investigate attack effectiveness when using alternate architectures. We explain two other architectures below.

StarGAN. StarGAN is an image-to-image translation model that improves over CycleGAN [76] by providing many-to-many domain translation capabilities. Unlike CycleGAN, StarGAN only requires a single generator and discriminator, but leverages an auxiliary domain classifier to ensure successful domain translation. StarGAN also includes a cycle loss term similar to CycleGAN to preserve content. We adapt StarGAN to fit into our *Jekyll* framework by adding a disease and identity loss term. To compute disease loss, we use the auxiliary domain classifier available in StarGAN, but use an external classifier for identity loss. Additionally, we observed that StarGAN's auxiliary classifier suffers significantly when classes are

imbalanced. To deal with such imbalance, we upsampled the underrepresented class, and replaced the binary-cross entropy loss used for the domain classification with focal loss [77]. More details are in Appendix C.

IPCGAN. Identity Preserving Conditional GAN or IPCGAN is an image translation GAN to synthesize face images in a targeted age group, while preserving identity, *e.g.,* translate a teenager's face image to one that looks 50+, while preserving identity. To achieve this, IPCGAN uses an age classifier to enforce translation to the new age group and implements identity loss as perceptual loss. However, IPCGAN lacks any kind of reconstruction/cycle loss, as used in *Jekyll*. To fit IPCGAN into *Jekyll* framework, we replace the age classifier by a disease classifier, and use our patient identity classifier. We do not add a cycle loss term to its training objective. More details are in Appendix C.

4. Experimental Setup for Evaluating Attack

To build *Jekyll*, we need to first train the disease and identity classifiers, followed by the GAN component (that uses the pre-trained disease and identity classifiers). We build two versions of each of the disease and identity classifiers. One version of classifiers are used to train the *Jekyll*, and are called the attack disease (C_a^d) and identity (C_a^i) classifiers. The other version of classifiers are used to evaluate the success of the attack, and are called the evaluation disease (C_e^d) and identity classifiers (C_e^i). Given images translated by *Jekyll*, the evaluation classifiers help in answering the following questions. (1) *Did we successfully inject the disease?* (2) *Did we preserve the identity?* We ensure that attack and evaluation classifiers are trained on datasets with no patient overlap.

For all attacks, we start with a victim set of healthy patients (i.e., having no diseases), and evaluate attack success by injecting different diseases (or different stages of same disease).

4.1. Medical Datasets

NIH chest X-ray dataset [52].[2] This is a publicly available dataset of 112,120 frontal chest X-ray images of 30,805 unique patients. Images are annotated with anonymized patient IDs, with labels indicating presence of one or more of 14 diseases. We demonstrate disease injection for two of these diseases, namely *Cardiomegaly*, and *Pleural Effusion*. Cardiomegaly causes an enlarged heart, usually the result of heart disease. Pleural Effusion is a condition that causes buildup of excess fluid around the lungs. These conditions are chosen because prior work demonstrated high detection accuracy for both using deep learning [17].

We partition the dataset by patients into two subsets. One partition is used for training *Jekyll*, including the GAN component, and the two attack classifiers (C_a^d, and C_a^i). The second partition is used only for evaluation which includes the victim set, and data for training the evaluation classifiers (C_e^d and C_e^i). The attack partition contains 24,000 patients, and the evaluation partition has 6,805 patients. Partitioning was performed in a manner that allows us to build reliable evaluation classifiers, *e.g.,*

---

1. This is because we use different types of transfer learning schemes.

2. https://nihcc.app.box.com/v/ChestXray-NIHCC

TABLE 1. # IMAGES USED TO TRAIN AND EVALUATE *Jekyll*.

Datasets	# Train images		# Victim images
	Disease	Non-disease	
Cardiomegaly	35,352	1,349	6,235
Effusion	35,352	4,977	6,883
Severe DR	32,728	1,000	680
Proliferative DR	32,728	982	703

TABLE 2. TESTING ACCURACIES OF THE DISEASE AND IDENTITY CLASSIFIERS USED.

Datasets	C_a^d	C_e^d	C_a^i	C_e^i
Cardiomegaly	84%	80%	98.2%	96.6%
Effusion	87%	80.7%	98.2%	96.6%
Severe DR	89.9%	90.4%	98.5%	99.9%
Proliferative DR	87%	87.1%	98.5%	99.9%

victim set should be large, and include patients with at least 10 images, so we could build a high quality identity classifier for evaluation (C_e^i). More details of data preparation are available in Appendix C.

The statistics of the data used to train and evaluate *Jekyll* are shown in Table 1 (see rows for Cardiomegaly and Effusion). Once trained, *Jekyll* is tested on over 6,000 victim images for both disease conditions. We also make sure that the victim set only includes patients with non-disease images (which can be then injected with a disease). Victim set includes images that correctly pass the disease classification test by C_e^d (as non-disease), and the identity classification test by C_e^i (as having the correct identity). This ensures that any effected style transfer is due to the success of *Jekyll*, and not due to misclassifications by the evaluation classifiers. Details of dataset (from attack and evaluation partition) used to train the attack and evaluation classifiers are in Tables 14 and 15 in Appendix C.

Retinal Fundus images. This is a publicly available dataset[3] provided by EyePACS, a platform for retinopathy screening. It consists of pairs (left and right eye) of retinal fundoscopy images for 88,702 patients. Images are annotated with anonymized patient IDs, with labels indicating different stages of Diabetic Retinopathy (DR)—no disease, mild, moderate, severe, and proliferative. DR is a disease impacting blood vessels in the retina leading to possible vision loss in people with diabetes. We demonstrate injection of severe and proliferative DR stages.

We prepare the dataset following a similar methodology as used for chest X-rays. Dataset statistics are shown in Table 1 (see rows for severe and proliferative DR). For both stages, our victim set includes over 600 images. Tables 14 and 15 in the Appendix show statistics of data used for the attack and evaluation (disease and identity) classifiers. More details are in Appendix C.

4.2. Training *Jekyll*, and evaluation classifiers

All models are implemented using Tensorflow v1.12.0 framework for Python.[4] An NVIDIA Titan Xp GPU with 12GB RAM, on a host with Intel(R) Xeon(R) W-2135 CPU @ 3.70GHz and 64 GB RAM was used for training. *Jekyll.* For all experiments, the Adam optimizer is used with learning rate of 0.0002, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate remains unchanged for the first 100 epochs and is then decreased linearly for the next 100 epochs. For each dataset, we empirically determine the weights for each of the loss terms. Using a validation set, we empirically estimate weights that produce the highest quality images, while ensuring successful injection

---

3. https://www.kaggle.com/c/diabetic-retinopathy-detection/data

4. Only exception is the StarGAN version of *Jekyll* which is implemented in PyTorch.

of disease and identity preservation. Training one instance of *Jekyll* takes ≈17 hours. Training configuration for each dataset and alternative architectures are in Appendix C.
Disease classifiers. For both datasets (X-ray and retinal), we leverage transfer learning. For the X-ray datasets, the teacher model is trained on relevant partitions (attack or evaluation depending on the classifier) of the NIH Chest X-ray dataset to diagnose all 14 available diseases (multi-label classifier), using the training setup used by Rajpurkar et al. [17]. To build our X-ray student model, we initialize our architecture (see earlier Section 3.2) with weights from the teacher model, and only fine-tune the last 70 layers. For the retinal DR disease classifiers, the teacher model is a DenseNet-121 architecture trained on ImageNet [75], and all layers are fine-tuned during training. Table 2 shows the accuracies of the (attack and evaluation) disease classifiers (C_a^d, and C_e^d) when applied to balanced test datasets. All classifiers have fairly high accuracy. Training configuration is available in Appendix C.
Identity classifiers. We again leverage transfer learning. For all datasets, the teacher model is a DenseNet-121 model trained on ImageNet. For each model, weights are initialized from the teacher model, and all layers are fine-tuned during training. The attack identity classifier, C_a^i, is trained to predict a random subset of patients in the *Jekyll* training dataset. It is not necessary to train the identity classifier on all patients in *Jekyll* training data, as we use a perceptual loss. For the retinal dataset, we perform data augmentation for both its training and testing data as we have limited data (only 2 images per patient). Blurring and random rotations are used to augment the dataset and create a set of 14 (including the original) images per patient. Table 2 shows the testing accuracies of the (attack and evaluation) identity classifiers (on balanced test datasets). All identity classifiers achieve over 96% accuracy. Training configuration is in Appendix C.

5. Evaluating Effectiveness of Attacks

We structure the evaluation of *Jekyll* based on our primary goals. More specifically, we aim to demonstrate that images generated by *Jekyll* show signs of disease and preserve patient identity. This requires misleading both real-life medical professional diagnostics, *as well as* machine learning classifiers that are used to aid diagnosticians. Therefore, we perform evaluation by: (1) different machine learning tools, and image quality metrics, and (2) by consulting medical professionals.

5.1. Evaluation by Machine Learning Tools and Image Quality Metrics

In this section, we examine different aspects of *Jekyll*'s effectiveness and design in detail. This includes evaluating: (1) image quality, (2) disease injection, (3) identity
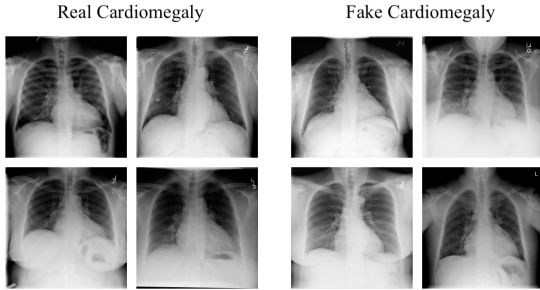
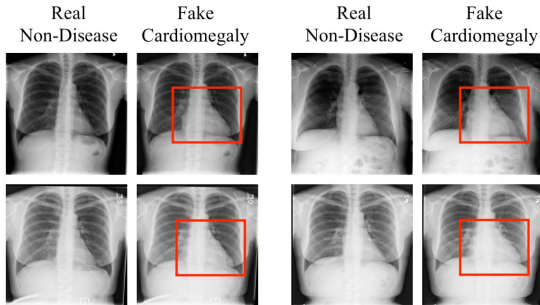Figure 2. Real and fake X-ray images with Cardiomegaly.

Figure 3. Injection of Cardiomegaly, red rectangles label the enlarged heart.

Figure 4. Real and fake X-ray images with Pleural Effusion.

Figure 5. Real and fake Retinal Fundus images with Proliferative DR.

preservation, (4) feasibility of sustaining the attack over time to match progression of disease, (5) effectiveness when using alternative architectures, and (6) design of *Jekyll*'s training objectives through an ablation study. For most of our evaluation, we focus on Cardiomegaly, and Effusion for X-rays, and proliferative DR for retinal images. We consider an earlier stage of DR—Severe DR, when evaluating progressive injection of disease.

5.1.1. Image Quality.

We start with analyzing quality of images generated by *Jekyll*. Image quality is important, otherwise the fake images can be easily caught by human inspection.

MSSIM. We use Multi-scale Structural Similarity (MSSIM) metric to evaluate image quality. MSSIM is a widely used objective image quality metric that correlates well with perceived image quality [78]. The metric is based on the idea that humans are sensitive to changes in structure, luminance, and contrast. This metric also considers perceived distortion from different viewing angles by analyzing the images at different scales. Given a reference image X, and a distorted version Y, $MSSIM(X, Y)$ ranges between 0 and 1, with 1 indicating that images are identical, and 0 indicating no structural similarity.

For each dataset, we first compute the MSSIM score between random pairs of real disease images, say between the sets Real A and Real B. Next, we compute MSSIM between images in real A and a random set of fake images with the same disease type/stage. We expect the MSSIM scores between real and fake images to be similar or higher than the scores between real images. If the MSSIM score between real and fake images is significantly lower than between real images, it would indicate that fake images look very different from real images of the same type (i.e., disease). Table 3 shows the average MSSIM scores,
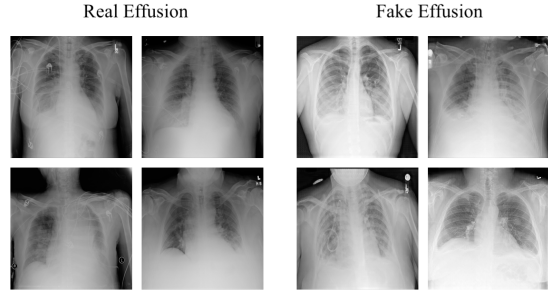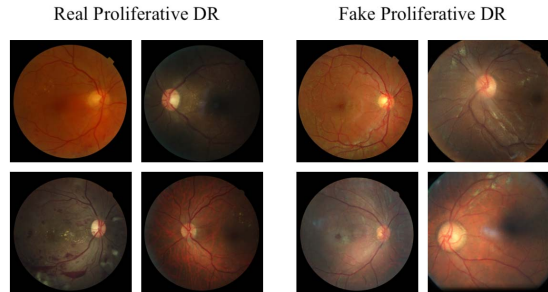
and the scores are within a similar range, suggesting that perceived image quality of fake images when compared to real images is satisfactory.

Figures 2, 4 and 5 show samples of fake and real images with Cardiomegaly, Effusion, and Proliferative DR, respectively. In these figures, for each fake image, a real image which is closest to the fake image in terms of L2 score is chosen. Showing such similar samples of real images helps to better understand image quality. Overall, we find that *Jekyll* is able to generate high quality images that are hard to distinguish from real images. More image samples can be found in Figure 15 in the Appendix.

FID and Inception score Inception score [78] and Fréchet Inception Distance [78] are popular quantitative metrics to evaluate quality of GAN generated images. Both rely on extracting embeddings from models trained on the ImageNet dataset. However, they are not suitable in our case, because the ImageNet dataset does not include biomedical images.

5.1.2. Disease Injection Success.

Disease injection rate. Success of disease injection is measured using the evaluation disease classifier, C_e^d. We compute a *disease injection rate*, R_d, as the percentage of generated images classified by C_e^d as having the disease. The rates are listed in the second column of Table 4. Disease injection rates are high for all diseases, with Proliferative DR showing 100% injection rate.

Figures 3, 6, and 7 show successful disease injections of Cardiomegaly, Effusion and Proliferative DR, respectively. The real non-disease image of the victim is shown next to each fake image. Changes can be observed in the fake images compared to the non-disease images. For Cardiomegaly in Figure 3, we see an enlarged heart (highlighted by red rectangle) compared to the non-disease image of the victim. *Jekyll* intelligently learned to selec-
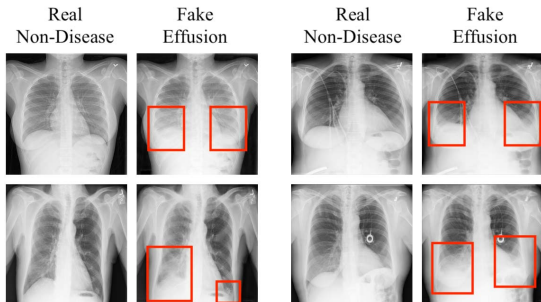
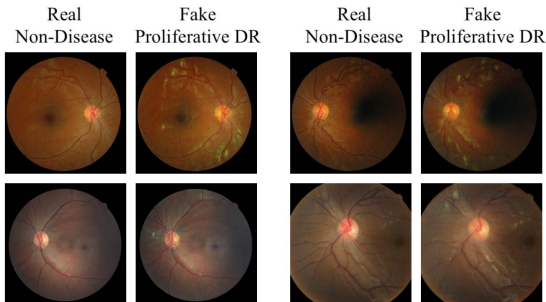Figure 6. Injection of Pleural Effusion, red rectangles label the fluid.

Figure 7. Injection of Proliferative DR. Cotton wool spots indicate disease.

Figure 8. Examples of failure cases for Cardiomegaly and Effusion.

TABLE 3. MSSIM SCORES TO EVALUATE IMAGE QUALITY OF FAKE IMAGES.

Datasets	MSSIM Score	
	Real A vs. Real B	Real A vs. Fake B
Cardiomegaly	0.456	0.449
Effusion	0.439	0.429
Proliferative DR	0.516	0.515

tively modify the heart region, while keeping rest of the image mostly similar. In Figure 6, one can observe build-up of excess fluid outside the lungs (marked by the red rectangles) as expected for Effusion. In the case of retinal images in Figure 7, there are some noticeable changes as well—*Jekyll* learned to generate "cotton wool"-like spots on the retina when DR is observed. This matches real description of symptoms for a patient with DR [79]. We provide more samples of disease injection in Figure 12 of the Appendix.

To better understand why we failed in some cases (for Cardiomegaly and Effusion), we manually went through some of the failed examples. Failed examples of Cardiomegaly are shown in Figure 8. We are able to identify two types of failures: *First*, there are cases where the generated image does show an enlarged heart, but maybe not large enough to pass the classifier test. *Second*, the original victim images did not have a distinct heart shaped region in the image, so it is harder for *Jekyll* to make modifications to that region. Failed examples of Pleural Effusion are shown in Figure 8. In failed examples, we observe only a *partial* buildup of fluid on one or both sides of lungs, which, while visible, is not significant enough to pass the classifier test. More failed samples are available in the Appendix (Figures 13, 14).

Disease classifier model interpretation. To further evaluate disease injection success, we leverage prior work on chest X-rays to identifying regions most indicative of a disease. Put simply, we check whether a machine learning model thinks we injected the disease in the right way, e.g., is the condition injected in the right place? Rajpurkar et al. [17] designed a model called ChexNet and present a method to identify regions of interest (for disease classification) in Chest X-rays by analyzing convolution feature maps. We use their tool and Figure 9 shows heat-
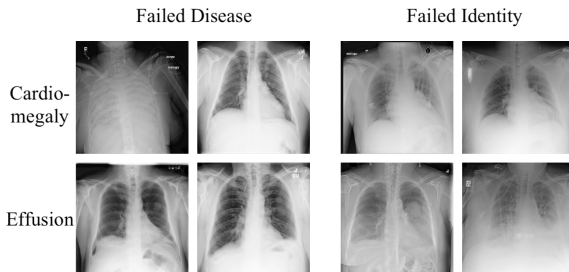
maps visualizing regions indicative of Cardiomegaly (red means higher probability) for both real and fake images. In both real and fake X-ray images with Cardiomegaly, we observe red regions highlighting the central and upper-right regions of the heart. In these regions, we observe visual enlargement of the heart, which is indicative of Cardiomegaly.

5.1.3. Identity Preservation Success.

Identity preservation rate. Similar to disease injection rate, we compute identity preserving rate, R_i using C_e^i. The percentage of generated images classified into the correct patient ID is the identity preserving rate. This is listed in Table 4, and is over 88% for all datasets, with X-ray modalities showing over 94% identity preservation rate. In Figures 3, 6, and 7, characteristics of generated images are quite similar to the victim's original image, except for any changes due to disease injection. The generated X-ray images presented in Figures 3, 6 indicate preservation of anatomical structure (positioning and shape of ribs, thoracic cavity, lungs, etc.), while generated retinal images in Figure 7 mostly preserve the vessel patterns as well.

We further investigated the samples that fail to preserve identity. Figures 8 shows failed examples of Cardiomegaly, and Effusion. Failed examples of Proliferative DR are in the Appendix (Figure 17). Overall, we suspect that injection of disease patterns disrupted other regions of the image that may have been crucial to capturing identity. For example, Effusion often introduces a fluid that can obscure large areas of the lung regions, hiding thoracic structure we believe to be crucial to patient identity. Similarly, in Proliferative DR, the 'cotton-wool' spots that arise due to insufficient blood supply, can often degrade the retinal structure associated with the patient.

Comparing retinal vessel masks. To further demonstrate identity preservation, we compare retinal vessel masks of the non-disease image and the generated image. The vascular patterns are known to aid in identification [80]. We thus aim to investigate the *similarity* between the vessel mask structure of the input non-disease image, and the generated image. If identity is preserved, we expect the vessel masks to be visually similar. To
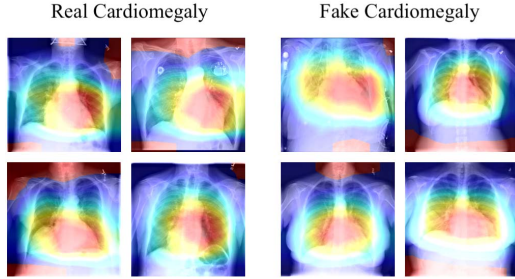
147

Real Cardiomegaly   Fake Cardiomegaly

Figure 9. Examples of heat-maps of real and fake cardiomegaly X-rays.

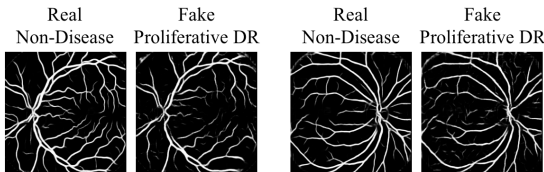Real Non-Disease   Fake Proliferative DR   Real Non-Disease   Fake Proliferative DR

Figure 10. Vessel segmentation masks of input non-disease and output Proliferative DR retinal images.

extract the vessel mask, we train a U-Net based vessel segmentation model [81] using DRIVE dataset [82]. Next, we apply the vessel extraction model to both the victim's original image, and the generated image. Paired examples are shown in Figure 10. We can see the vessel structure is mostly preserved in the generated images. To obtain a quantitative measure, we compute MSSIM scores between the input and output vessel structures. For Proliferative DR, we obtain an MSSIM score of 0.71. Score is not a perfect 1.0, because blood vessels can swell and leak in the case of Proliferative DR. We also examined MSSIM scores for Severe DR, and obtain higher scores of 0.89, which is expected as this is a less advanced stage.

5.1.4. Sustaining the Attack Over Time.

One of our attack goals is to enable repeated attacks using *Jekyll* over time. We consider a setting where attacker wants to mimic the natural progression of a disease condition, *e.g.,* condition worsening over time. As discussed in Section 3.2, we consider two scenarios.

Disease stage data available. We demonstrate the efficacy of this approach using the retinal dataset. In addition to Proliferative DR, we demonstrate injection of an earlier stage, namely Severe DR. Disease injection (R_d) and identity preservation (R_i) rates for this stage is presented in Table 5. *Jekyll* is able to achieve over 99% disease injection rate, and over 88% identity preservation rate for both Proliferative and Severe stages. Sample images are provided in Appendix A (Figure 16).

Disease stage data unavailable. In this case, attacker has no data to train *Jekyll* to target different stages. We use linear interpolation (Section 3.2) to produce intermediate stages of Cardiomegaly. Figure 11 illustrates one example, where we produce 3 intermediate stages of Cardiomegaly using degree of disease injection, $\alpha \in [0.25, 0.50, 0.75]$. We see that this simulates enlargement of heart over time, while preserving patient identity. We further verified several such examples (using different victim images) with a doctor experienced in analyzing chest X-rays (who also participated in our user study in Section 5.2). The

TABLE 4. DISEASE INJECTION (R_d) AND IDENTITY PRESERVING (R_i) PERFORMANCE OF *Jekyll*.

Dataset	R_d	R_i
Cardiomegaly	82.8%	95.2%
Effusion	95.7%	94.4%
Proliferative DR	100%	88.4%

TABLE 5. DISEASE INJECTION (R_d) AND IDENTITY PRESERVING (R_i) PERFORMANCE OF PROGRESSIVELY INJECTING A DISEASE.

Dataset	R_d	R_i
Severe DR	99.6%	97.5%
Proliferative DR	100%	88.4%

doctor confirmed that these intermediate stages are indeed plausible. To obtain a quantitative measure for progressive disease injection, we further apply the evaluation disease classifier, C_e^d on all intermediate stages of all victims. The classifier produces a monotonically increasing probability of image being in the Cardiomegaly class, when we increase α (as shown in Figure 18 in Appendix A).

Overall, in both settings, *Jekyll* is able to mimic progression of a disease.

5.1.5. Effectiveness of Jekyll when using alternative architectures.
We evaluate alternative neural network architectures to implement *Jekyll*. Two existing image-to-image translation models, StarGAN and IPCGAN are adapted to fit into *Jekyll*'s framework (discussed earlier in Section 3.2). In addition, as a baseline, we evaluate how a basic image-to-image translation framework would perform on our attack dataset. For this baseline, we choose the Pix2Pix [72] model, an image-to-image translation GAN that requires a paired dataset. Training objectives for Pix2Pix include an adversarial loss term, and an $L1$ loss term that minimizes the differences between the generated image and the targeted image in the pair. There are no other loss terms to enforce disease injection or identity. Since we do not have paired data, we use random pairs of non-disease and disease image to train Pix2Pix.

Table 6 presents disease injection and identity preservation rates for all architectures—StarGAN, IPCGAN and Pix2Pix. We note that all architectures produce high quality images. However, there is variance in their attack effectiveness, and our original architecture outperforms all of them. *StarGAN* performs reasonably well, achieving over 70%, and 79%, disease injection and identity preservation rate, respectively for all datasets. We suspect that using a single generator-discriminator pair in StarGAN (compared to using 2 pairs in our framework) is negatively impacting its translation performance. On the other hand, *IPCGAN* achieves high identity preservation rate of over 99% for all datasets, but disease injection rates are lower, with the retinal dataset showing only over 2% performance. While adjusting the weights of the loss terms, we observe that it is necessary to upweight the identity loss term in order to obtain passable image quality. This likely explains the high R_i rate. Any attempt to raise the weight of the disease loss term resulted in lowered image quality. Also note that IPCGAN does not include a cycle loss term. Lastly, *Pix2Pix* exhibits high disease injection performance, but almost completely fails to preserve identity. This is expected given that Pix2Pix expects paired data (which is

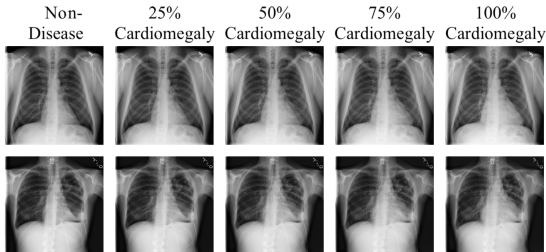| Datasets	Jekyll		Only L_{cycle}		$L_{disease}$ removed		$L_{identity}$ removed		Pix2Pix		IPCGAN		StarGAN	
	R_d	R_i	R_d	R_i	R_d	R_i	R_d	R_i	R_d	R_i	R_d	R_i	R_d	R_i
Cardiomegaly	82.8%	95.2%	45.2%	94.2%	49.4%	94.6%	75.7%	86.7%	62.2%	0.3%	36.1%	99.3%	74.6%	79.8%
Effusion	95.7%	94.4%	44.2%	96.9%	55.2%	97.1%	97.5%	91.4%	97.6%	0.08%	64.3%	99.3%	70%	90.8%
Proliferative DR	100%	88.4%	96.7%	85%	89.3%	93.8%	99.8%	82.2%	100%	0.1%	2.4%	98.5%	87.9%	93.1%

Figure 11. Examples of progressive injection for Cardiomegaly using linear interpolation.

not available) and includes no train objectives to preserve identity (cycle loss or identity loss).

5.1.6. Ablation Study. Table 6 presents results of our ablation study. In all experiments, we retain the adversarial loss and cycle loss (L_{cycle}) terms, as they are basic building blocks of *Jekyll*'s training objectives. On removing the disease loss term ($L_{disease}$), disease injection rates decrease for all three datasets, with X-ray datasets exhibiting R_d below 55%. Hence, the adversarial loss term on its own is insufficient for effective disease injection all the time, and we need a disease loss term. However, since disease-specific perturbations are no longer enforced, the patients' structural integrity is easily preserved, and the identity preservation rate (R_i) is high for all diseases.

To preserve identity, *Jekyll* relies on the cycle loss (L_{cycle}) and identity loss ($L_{identity}$) terms. When $L_{identity}$ is removed, R_i decreases for all three diseases—an average of 5.8% decrease for chest diseases, and 6.2% decrease for Proliferative DR. However, the cycle loss term helps to avoid significant drop in identity preservation rate. We also observe noise artifacts, along with blurring in images generated without the identity loss. Therefore, the perceptual nature of identity loss contributes to image quality as well.

When both $L_{disease}$, and $L_{identity}$ are removed, *Jekyll* only uses the cycle loss term, and is equivalent to the vanilla CycleGAN translation model [73]. The disease injection rate drops significantly for both X-ray datasets (below 46%), while R_i is high across all datasets. This is because when disease perturbations are not enforced, it is easy to achieve high R_i. This shows that a vanilla CycleGAN model is not sufficient for our attack.

To summarize, $L_{disease}$ is crucial to achieve high disease injection rate. But substantial disease perturbations can hurt the identity of the patient, and therefore $L_{identity}$ helps to balance things out, and preserve identity.

5.2. Evaluation by Medical Professionals

To understand *Jekyll*'s ability to mislead medical professionals, we recruit medical professionals experienced in Chest X-ray diagnostics and conduct a user study. We investigate two key questions: (1) *Can we convince medical professionals that* Jekyll *generated images contain the targeted disease condition?* (2) *Can medical professionals distinguish between real images and* Jekyll *generated (fake) images?* We use the Chest X-ray dataset, and Cardiomegaly as the attacker targeted disease condition. Due to logistical difficulty in recruiting professionals with adequate experience in assessing retinal fundus imagery, we limit the study to the Chest X-ray dataset. Note that our study does not impact any real patients. Prior to conducting our study, we submitted a human subject protocol and received approval from our local IRB board.

We recruited three qualified medical practitioners, with extensive experience in evaluating chest X-Rays. Two evaluators are resident physicians in the Internal Medicine Department at a hospital. The third evaluator is a senior resident physician in the General Medicine Department. All three evaluators work at different hospitals. A pre-study questionnaire confirmed that all three practitioners have experience with analyzing chest X-rays.

The study requires both fake images with disease condition, and real images with and without disease condition. We sampled disease images (real and fake) from those having a disease probability higher than 0.8 when evaluated by the evaluation disease classifier (C_e^d). Further, we observe that real X-rays contain watermarks (usually in upper right corner). *Jekyll* images contained slightly blurry reproduction of watermarks in the original victim's image. To ensure a fair evaluation, where evaluators would focus on the biomedical content, rather than quality of watermarks, we apply a simple automated post-processing step to sharpen the watermarks in fake images. [5]

5.2.1. User Study Part 1: Evaluating Disease Injection. In this task, given an X-ray image, evaluator has to determine whether it contains the disease condition of Cardiomegaly. Evaluators are offered three choices: (1) 'Disease', image indicates the disease condition, (2) 'No disease', no disease condition is observed, and (3) 'Other', if evaluator is unable to choose (1) or (2) for some reason. We did not enforce a time limit to evaluate a given image. Figure 19 in Appendix B shows our survey page.

Each evaluator is shown a total of 150 images, out of which 100 are real, and the remaining 50 are fake images containing Cardiomegaly. Among the 100 real images, 50 contain no disease condition, and the other 50 contain Cardiomegaly. Evaluators are not aware of the presence of fake images. Each of the three evaluators independently evaluated all 150 images, in a total of 5 sessions (30 images per session), providing us a total

---

5. We use a similar approach from Section 3.2 (on sustaining attack over time) and apply a linear interpolation scheme in a localized region containing the watermark, using the original non-disease image and the generated image.

of 450 judgements. Out of these 450 judgements, we discarded 5 where the evaluators were uncertain about their decision ('Other' class).[6]

We start by investigating the baseline accuracy of the evaluators for this task, by comparing their judgments on the 100 real images. Second column of Table 7 shows the individual accuracy of evaluators, as well as the average and majority accuracy. The average accuracy is computed by taking the mean of the individual accuracies, while majority accuracy is computed by taking the majority judgments of the three evaluators. On average, accuracy of the evaluators is 0.868, while when taking majority judgments, the accuracy rises to 0.909. Corresponding false-positive and false-negative rates are provided in Tables 10 and 11 (in Appendix). The inter-annotator agreement, when computed using Fleiss' κ metric, is $\kappa = 0.62$ ($p < 10^{-6}$) indicating substantial agreement.

Next, we analyze the judgments of evaluators on the 50 fake images (with Cardiomegaly) to understand the ability of *Jekyll* in successfully injecting realistic looking disease conditions. Column 3 of Table 8 shows the percentage of fake images that were judged to contain diseased symptoms by zero, at least one, at least two or by all three evaluators. We can see that for 60% of the cases all evaluators agreed that the fake images contained symptoms of the disease, while in overall 80% cases, the majority of the evaluators agreed that the fake images contained symptoms of the disease.

Finally, we analyze if there is any statistical difference between judgements of real and fake disease images. Second column of Table 8 also shows the percentage of real images that were judged to contain disease symptoms by zero or more evaluators. To understand whether this represented statistically significant differences, we compared the ratio of the number of disease vs non-disease judged images (according to the majority) using Pearson's Chi-squared test (a parametric test) and Fisher's Exact Test (a non-parametric test). When comparing the differences using Pearson's Chi-squared test we obtain $\chi^2 = 0.669$ ($p = 0.4132$). Similarly, when using Fisher's Exact Test, we obtain an odds ratio of 1.822 ($p = 0.4139$). Since in both cases $p >> 0.05$ we do not consider the differences to be statistically significant.

To summarize, medical professionals are convinced that *Jekyll* images contain the targeted disease (Cardiomegaly), and their judgements of real and fake disease images shows no statistical difference.

5.2.2. User Study Part 2: Detecting Fake Images. In this task, evaluators are asked to mark a given image as 'fake' or 'real'. We described that 'real' images are those produced by an X-ray imaging device, and 'fake' are those generated by a computer algorithm. Evaluators were free to use any basis for their judgement. Again, we do not enforce any time limit to analyze the image. Figure 20 in Appendix B shows a screenshot of our survey page.

All three evaluators are given 50 real images with disease, and 50 fake images (with disease) generated using *Jekyll*. Third column of Table 7 shows the individual accuracy of evaluators, as well as the average, and majority

6. For these 5 uncertain cases, evaluators gave us feedback that some images were rotated, or had poor visibility in the heart region, making it harder to arrive at a decision.

Evaluator	Accuracy	
	Study Part 1 (Real X-rays only)	Study Part 2
Evaluator 1	91.9%	48.0%
Evaluator 2	86.6%	48.0%
Evaluator 3	91.8%	55.0%
Average	86.8%	50.3%
Majority	90.9%	47.0%

TABLE 8. PERCENTAGE OF REAL X-RAY IMAGES AND FAKE X-RAY IMAGES JUDGED BY EVALUATORS TO CONTAIN SYMPTOMS OF CARDIOMEGALY.

# Evaluators voting 'Disease'	Real Images	Fake Images
At least 1 evaluator	98%	98%
At least 2 evaluators	88%	80%
All 3 evaluators	60%	60%
0 evaluators	2%	2%

accuracy. Overall, the accuracy when considering majority decision is low at 47%. The inter annotator agreement for the fake detection task is $\kappa = 0.024$ ($p = 0.674$), when measured using Fleiss's κ metric, indicating a lack of agreement among annotators. Therefore, we conclude that evaluators are unable to accurately distinguish between real and fake images.

6. Defending Against Attacks by *Jekyll*

We investigate two methods to detect images generated by *Jekyll*: (1) *blind detection*, and (2) *supervised detection*

Blind Detection. In this scheme, the defender has no access to fake images and no knowledge of the attacker's generative model. However, a corpus of real images is available to the defender. These assumptions are realistic, but the problem setting is quite challenging. A viable defense approach in this scenario is to use anomaly detection to identify fake images.

There is limited existing work on blind detection of GAN generated images. We use a method proposed by Li et al. [83] that leverages disparities in color components between real and fake images (GAN generated images). The idea is that GAN training objectives typically place no explicit constrains to learn specific correlations among color components in the RGB space. This can result in inconsistencies when fake images are examined in other color spaces, namely HSV and YCbCr. This technique extracts color statistics of HSV and YCbCr color spaces of real images, and then uses these features to train a One-Class Support Vector Machine (SVM) for anomaly detection. The expectation is that the one-class SVM will flag fake images as anomalies in the (color) feature space. We refer to this method as *CSD-SVM*.

Since this method requires color images, we only use the Proliferative DR dataset. X-ray images are in grayscale. We use an implementation of CSD-SVM provided by the authors.[7] Training dataset of CSD-SVM includes 232 real images with Proliferative DR. The testing dataset contains 700 real, and 703 fake images (both with

7. https://github.com/lihaod/GAN_image_detection

TABLE 9. PERFORMANCE OF THE SUPERVISED MESONET CLASSIFIER AND BLIND CSD SVM WHEN DETECTING FAKE IMAGES GENERATED BY *Jekyll*. PRECISION AND RECALL ARE PRESENTED FOR THE FAKE IMAGES GENERATED BY *Jekyll*.

| Datasets | Supervised: MesoNet | | | | | | Blind: CSD-SVM | | |
| | Jekyll Images | | | Repurposed Jekyll Images | | | Jekyll Images | | |
	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Cardiomegaly	99.4%	100%	99%	56.8%	100%	6%	N/A	N/A	N/A
Proliferative DR	98.1%	97%	99%	83.4%	96%	70%	60.1%	29%	77%

Proliferative DR). The SVM uses a radial basis kernel, where kernel coefficient (γ) is set to 0.1, and the upper bound for the training error (ν) is set to 0.12. The hyper-parameters are estimated using 5-fold cross validation.

Table 9 shows detection accuracy, precision, and recall (for fake class). CSD-SVM is only able to achieve 60% accuracy, at a precision and recall of 29%, and 77%, respectively. The low performance suggests that images generated by *Jekyll* are hard to detect using a blind scheme that analyzes color space disparity.

Supervised Detection. Here we assume the defender has access to a corpus of both real and fake images. While this is a stronger defense assumption, it is important to investigate how well supervised schemes can detect fake images. Such schemes can be used along with a blind detection scheme to further improve detection performance. The basic idea is to build a supervised machine learning classifier to detect fake images.

We use a high performing supervised classifier called MesoNet [84], proposed recently by Afchar et al. The intuition behind *MesoNet* is to examine image features at a *mesoscopic* level, as micro (image noise) and macroscopic (semantic aspects) features may fail to distinguish between real and fake imagery. We use an implementation of MesoNet provided by the authors.[8]. The MesoNet model includes 4 inception blocks, where each block concatenates several convolutional layers of varying shapes together. Training and testing sets were constructed on disjoint sets of patients.

We apply MesoNet to the Cardiomegaly and Proliferative DR datasets. For the Cardiomegaly dataset, we train MesoNet on $4,920$ real (obtained via upsampling), and $4,359$ fake images, and test on a set of 443 real and 375 fake images. For the Proliferative DR dataset, we train MesoNet on 400 real and 403 fake images, and test on a balanced set of 300 real and 300 fake images.

Table 9 shows the detection performance of MesoNet. MesoNet performs well in detecting images generated by *Jekyll* with an accuracy of 99.4% on the Cardiomegaly dataset, and 98.1% accuracy on the Proliferative DR dataset. This is promising. However, note that the attacker can try to adapt the *Jekyll* model to bypass detection. Next, we present a countermeasure against this defense.

Countermeasures Against Supervised Detection. Our idea is to modify the attacker's GAN model to evade detection. If the attacker has white-box access to the defender model, one evasion strategy is to attach the defender's model to *Jekyll* as an additional discriminator. This will train the generator to produce fake images that can fool the defender's model. Instead, we assume a more

realistic setting that does not require access to the defender model.

We tested different modifications to *Jekyll*, including retraining with different random seeds, retraining with different weights for the loss terms, changing the generator architecture by varying configuration of the deconvolution filters, and finally modifying the discriminator framework. Among all these modifications, we observe that changes to the discriminator framework results in successful evasion, which we describe in more detail below.

In a GAN, the generator primarily relies on the discriminator's feedback to generate fake images. *Jekyll* uses a PatchGAN discriminator that determines whether individual small patches of an image look real or fake. We modify *Jekyll* to include an additional discriminator that analyzes the image as a whole (instead of individual patches) to classify as fake or real. This additional discriminator is implemented using a single layer feed-forward network that takes the output of PatchGAN and produces a scalar score. The modified adversarial GAN objective is then updated to include both the direct PatchGAN output, as well as the output of the feed-forward network. Such an additional discriminator serves to approximate the role of a defender model, which *Jekyll*'s generator can learn to fool while training. We refer to this modified version of *Jekyll* as *Repurposed Jekyll*.

We then apply MesoNet trained on *Jekyll* images to images generated by *Repurposed Jekyll*. Table 9 shows the detection performance. There is a significant drop in detection accuracy from 99.4% to 55.6% for the Cardiomegaly dataset, and from 98.1% to 83.4% for the Proliferative DR dataset. These results highlight the susceptibility of supervised classifiers to advanced attacks.

7. Conclusion

In this work, we investigate deep learning powered attacks against medical image diagnostics. We present *Jekyll*, a GAN-based framework that can translate a biomedical image of a patient to a new one that indicates an attacker chosen disease condition, while preserving the identity of the patient. Such translation attacks can lead to misdiagnosis by both medical professionals and machine learning algorithms. Additionally, *Jekyll* provides methods to enable repeated attacks against a victim by controlling the severity of the injected disease. We extensively evaluate attack success of *Jekyll* using both machine learning tools and a user study involving medical professionals. Lastly, we investigate defensive measures that aim to detect *Jekyll* generated images. We find that supervised detection approaches are promising but vulnerable to evasion tactics by advanced attackers. We hope our work encourages the community to pursue robust defensive measures.

8. https://github.com/DariusAf/MesoNet

References

[1] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar, H. S. Anderson, H. Roff, G. C. Allen, J. Steinhardt, C. Flynn, S. Ó. hÉigeartaigh, S. Beard, H. Belfield, S. Farquhar, C. Lyle, R. Crootof, O. Evans, M. Page, J. Bryson, R. Yampolskiy, and D. Amodei, "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," *CoRR abs/1802.07228*, 2018.

[2] D. O'Sullivan, "When seeing is no longer believing." 2019. [Online]. Available: https://www.cnn.com/interactive/2019/01/business/pentagons-race-against-deepfakes/

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. of NIPS*, 2014.

[4] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *Proc. of ICLR*, 2018.

[5] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. of CVPR*, 2018.

[6] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proc. of NeurIPS*, 2016.

[7] N. Christopher, "We've just seen the first use of deepfakes in an indian election campaign," 2020. [Online]. Available: https://www.vice.com/en_in/article/jgedjb/the-first-use-of-deepfakes-in-indian-election-by-bjp

[8] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, pp. 1287–1289, 2019.

[9] A. Al-Mawali, J. Idikula, and A. D. Pinto, "Fraud and misconduct in clinical research: a step to improve ethical practice in research," *Journal of Contemporary Medical Sciences*, vol. 4, 2018.

[10] L. Morris, "Combating fraud in health care: an essential component of any cost containment strategy," *Health Affairs*, vol. 28, pp. 1351–1356, 2009.

[11] A. Rashidian, H. Joudaki, and T. Vian, "No evidence of the effect of the interventions to combat health care fraud and abuse: a systematic review of literature," *PloS one*, vol. 7, 2012.

[12] R. A. Bauder and T. M. Khoshgoftaar, "Medicare fraud detection using machine learning methods," in *Proc. of ICMLA*, 2017.

[13] M. Herland, T. M. Khoshgoftaar, and R. A. Bauder, "Big data fraud detection using multiple medicare data sources," *Journal of Big Data*, vol. 5, p. 29, 2018.

[14] C. Beek, "Mcafee researchers find poor security exposes medical data to cybercriminals," Mar 2018. [Online]. Available: https://www.mcafee.com/blogs/other-blogs/other-blogs/mcafee-labs/mcafee-researchers-find-poor-security-exposes-medical-data-to-cybercriminals/

[15] Verizon, "Protected health information data breach report," 2018. [Online]. Available: https://enterprise.verizon.com/resources/reports/protected_health_information_data_breach_report.pdf

[16] Y. Mirsky, T. Mahler, I. Shelef, and Y. Elovici, "Ct-gan: Malicious tampering of 3d medical imagery using deep learning," in *Proc. of USENIX Security Symposium*, 2019.

[17] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *CoRR abs/1711.05225*, 2017.

[18] C. Osborne, "ibm-takes-on-alzheimers-disease-with-machine-learning," 2019. [Online]. Available: https://www.zdnet.com/article/ibm-takes-on-alzheimers-disease-with-machine-learning/

[19] T. Jo, K. Nho, and A. J. Saykin, "Deep learning in alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data," *Frontiers in aging neuroscience*, vol. 11, p. 220, 2019.

[20] M. Rushanan, D. Foo Kune, C. M. Swanson, and A. D. Rubin, "SoK: Security and Privacy in Implantable Medical Devices and Body Area Networks," in *Proc. of IEEE S&P*, 2014.

[21] B. Sawyer and C. Cox, "How does health spending in the U.S. compare to other countries?" 2018. [Online]. Available: https://www.healthsystemtracker.org/chart-collection/health-spending-u-s-compare-countries/

[22] J. E. Wennberg, A. M. O'Connor, E. D. Collins, and J. N. Weinstein, "Extending the p4p agenda, part 1: how medicare can improve patient decision making and reduce unnecessary care," *Health Affairs*, vol. 26, pp. 1564–1574, 2007.

[23] H. Joudaki, A. Rashidian, B. Minaei-Bidgoli, M. Mahmoodi, B. Geraili, M. Nasiri, and M. Arab, "Using data mining to detect health care fraud and abuse: a review of literature," *Global journal of health science*, vol. 7, p. 194, 2015.

[24] D. L. Barlett and J. B. Steele, *Critical condition: how health care in America became big business-and bad medicine*. Broadway, 2006.

[25] C. K. Reddy and C. C. Aggarwal, *Healthcare data analytics*. Chapman and Hall/CRC, 2015.

[26] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *Jama*, vol. 316, pp. 2402–2410, 2016.

[27] G. Simon, "Principles of chest x-ray diagnosis," *American Journal of Physical Medicine & Rehabilitation*, vol. 51, p. 42, 1972.

[28] "Artificial intelligence that reads chest x-rays is approved by fda," Nov 2019. [Online]. Available: https://www.ucsf.edu/news/2019/09/415406/artificial-intelligence-reads-chest-x-rays-approved-fda

[29] C. Metz, "India fights diabetic blindness with help from a.i." 2019. [Online]. Available: https://www.nytimes.com/2019/03/10/technology/artificial-intelligence-eye-hospital-india.html

[30] J. SNOW, "The algorithm will see you now: How ai is helping doctors diagnose and treat patients," 2019. [Online]. Available: https://www.pbs.org/wgbh/nova/article/how-ai-is-helping-doctors-diagnose-and-treat-patients/

[31] B. Siwicki, "Johns Hopkins researchers use deep learning to combat pancreatic cancer," 2018. [Online]. Available: https://www.healthcareitnews.com/news/johns-hopkins-researchers-use-deep-learning-combat-pancreatic-cancer

[32] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," *CoRR abs/1606.05718*, 2016.

[33] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115–118, 2017.

[34] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature medicine*, vol. 25, p. 65, 2019.

[35] M. J. Van Grinsven, B. van Ginneken, C. B. Hoyng, T. Theelen, and C. I. Sánchez, "Fast convolutional neural network training using selective data sampling: Application to hemorrhage detection in color fundus images," *IEEE transactions on medical imaging*, vol. 35, pp. 1273–1284, 2016.

[36] R. Rajalakshmi, R. Subashini, R. M. Anjana, and V. Mohan, "Automated diabetic retinopathy detection in smartphone-based fundus photography using artificial intelligence," *Eye*, vol. 32, pp. 1138–1144, 2018.

[37] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang, "Deepsec: A uniform platform for security analysis of deep learning model," in *Proc. of IEEE S&P*, 2019.

[38] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, pp. 2805–2824, 2019.

[39] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Proc. of ICLR*, 2014.

[40] M. Juuti, B. Sun, T. Mori, and N. Asokan, "Stay on-topic: Generating context-specific fake restaurant reviews," in *Proc. of ESORICS*, 2018.

[41] Y. Yao, B. Viswanath, J. Cryan, H. Zheng, and B. Y. Zhao, "Automated crowdturfing attacks and defenses in online review systems," in *Proc. of CCS*, 2017.

[42] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. of ICLR*, 2015.

[43] A. Das and R. Verma, "Automated email generation for targeted attacks using natural language," *CoRR abs/1908.06893*, 2019.

[44] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *Proc. of USENIX Security*, 2018.

[45] D. Iter, J. Huang, and M. Jermann, "Generating adversarial examples for speech recognition," *Stanford Technical Report*, 2017.

[46] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proc. of CCS*, 2017.

[47] M. Nasr, A. Bahramali, and A. Houmansadr, "Deepcorr: Strong flow correlation attacks on tor using deep learning," in *Proc. of CCS*, 2018.

[48] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal, "Raptor: Routing attacks on privacy in tor," in *Proc. of USENIX Security*, 2015.

[49] Q. Yan, M. Wang, W. Huang, X. Luo, and F. R. Yu, "Automatically synthesizing dos attack traces using generative adversarial networks," *IJMLC*, vol. 10, p. 3387–3396, 2019.

[50] N. Kohli, D. Yadav, M. Vatsa, R. Singh, and A. Noore, "Synthetic iris presentation attack using idcgan," in *Proc. of IJCB*, 2017.

[51] R. H. Choplin, J. M. Boehme, and C. D. Maynard, "Picture archiving and communication systems: an overview." *RadioGraphics*, vol. 12, pp. 127–129, 1992.

[52] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proc. of CVPR*, 2017.

[53] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (oasis): Cross-sectional mri data in young, diddle aged, non-demented, and demented older adults," *J. Cognitive Neuroscience*, vol. 12, pp. 2677–2684, 2007.

[54] K. Yan, X. Wang, L. Lu, and R. M. Summers, "Deeplesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning," *Journal of medical imaging*, vol. 5, p. 036501, 2018.

[55] S. G. Armato III, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, L. P. Clarke *et al.*, "Data from LIDC-IDRI. the cancer imaging archive," 2015. [Online]. Available: http://doi.org/10.7937

[56] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. of ICLR*, 2015.

[57] S. Qamber, Z. Waheed, and M. U. Akram, "Personal identification system based on vascular pattern of human retina," in *Proc. of CIBEC*, 2012.

[58] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. of ICLR*, 2019.

[59] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," in *Proc. of IEEE ISBI*, 2018.

[60] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *Proc. of IWSSMI*, 2018.

[61] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[62] A. Madani, M. Moradi, A. Karargyris, and T. Syeda-Mahmood, "Chest x-ray generation and data augmentation for cardiovascular abnormality classification," in *Proc. of SPIE Medical Imaging*, 2018.

[63] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. of IPMI*, 2017.

[64] J. Son, S. J. Park, and K.-H. Jung, "Retinal vessel segmentation in fundoscopic images with generative adversarial networks," *CoRR abs/1706.09318*, 2017.

[65] C. F. Baumgartner, L. M. Koch, K. Can Tezcan, J. Xi Ang, and E. Konukoglu, "Visual feature attribution using wasserstein gans," in *Proc. of CVPR*, 2018.

[66] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Medical image analysis*, vol. 58, p. 101552, 2019.

[67] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, pp. 383–392, 2018.

[68] P. Moeskops, M. Veta, M. W. Lafarge, K. A. Eppenhof, and J. P. Pluim, "Adversarial training and dilated convolutions for brain mri segmentation," in *Proc. of MICCAI*, 2017.

[69] D. Yang, D. Xu, S. K. Zhou, B. Georgescu, M. Chen, S. Grbic, D. Metaxas, and D. Comaniciu, "Automatic liver segmentation using an adversarial image-to-image network," in *Proc. of MICCAI*, 2017.

[70] Z. Li, Y. Wang, and J. Yu, "Brain tumor segmentation using an adversarial network," in *Proc. of International MICCAI Brainlesion Workshop*, 2017.

[71] D. Jin, Z. Xu, Y. Tang, A. P. Harrison, and D. J. Mollura, "CT-realistic lung nodule simulation from 3D conditional generative adversarial networks for robust lung segmentation," in *Proc. of MICCAI*, 2018.

[72] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. of CVPR*, 2017.

[73] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. of ICCV*, 2017.

[74] X. Hu, "CycleGAN-tensorflow," 2018. [Online]. Available: https://github.com/xhujoy/CycleGAN-tensorflow

[75] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of CVPR*, 2017.

[76] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Star-GAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. of CVPR*, 2018.

[77] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. of ICCV*, 2017.

[78] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[79] AOA, "American optometric association," 2018. [Online]. Available: https://www.aoa.org/patients-and-public/eye-and-vision-problems/glossary-of-eye-and-vision-conditions/diabetic-retinopathy

[80] P. Costa, A. Galdran, M. Meyer, M. Abramoff, M. Niemeijer, A. Mendonca, and A. Campilho, "Towards adversarial retinal image synthesis," *CoRR abs/1701.08974*, 2017.

[81] Orobix, "Retina blood vessel segmentation with a convolutional neural network," 2018. [Online]. Available: https://github.com/orobix/retina-unet

[82] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 501–509, 2004.

[83] H. Li, B. Li, S. Tan, and J. Huang, "Detection of deep network generated images using disparities in color components," *CoRR abs/1808.07276*, 2018.

[84] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: a compact facial video forgery detection network," in *Proc. of WIFS*, 2018.

Appendix A.
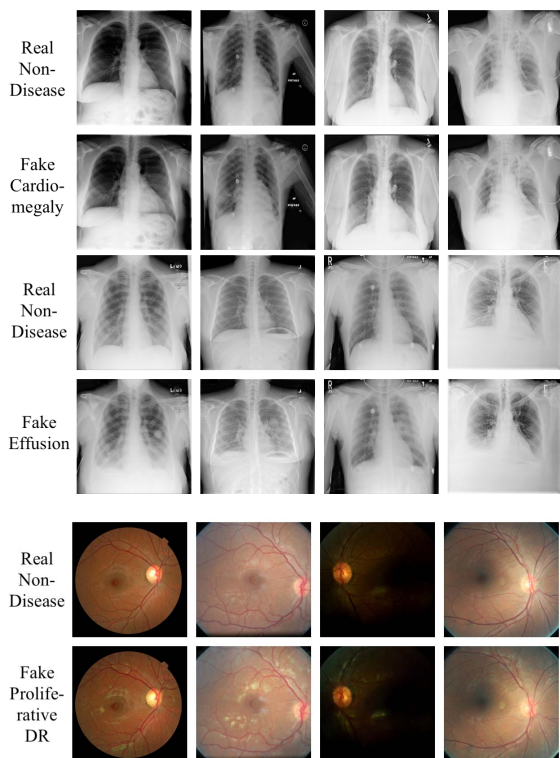## Supplementary Images and Figures

Figure 12. Disease Injection Samples for Cardiomegaly, Effusion and Proliferative DR
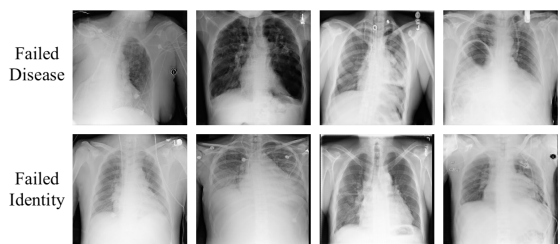

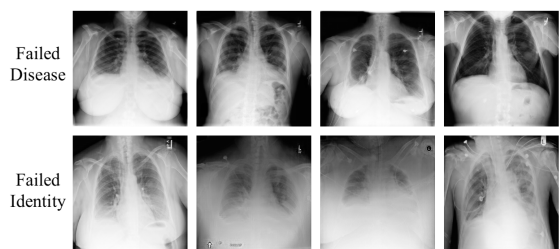Figure 13. Failed Cardiomegaly Samples.
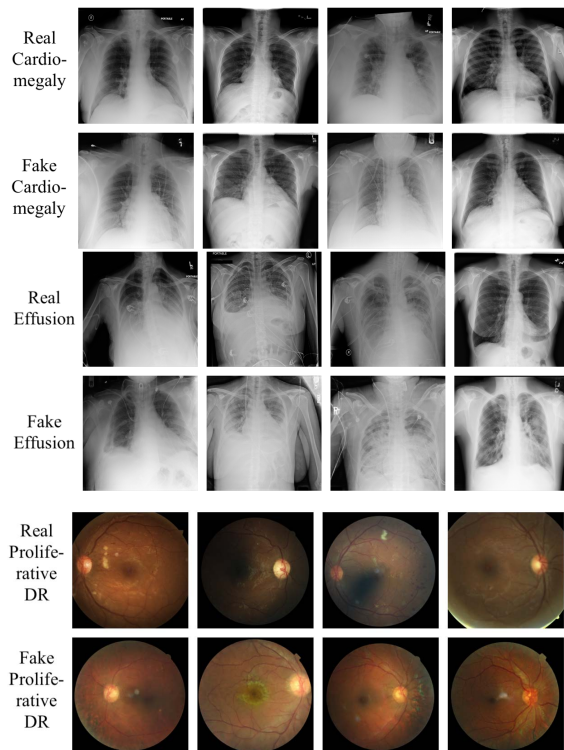

Figure 14. Failed Effusion Samples.

Figure 15. Real and Fake Samples for Cardiomegaly, Effusion and Proliferative DR
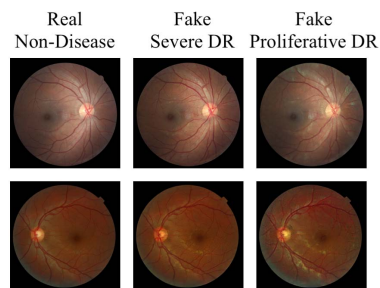

Figure 16. Examples of progressive injection for DR.

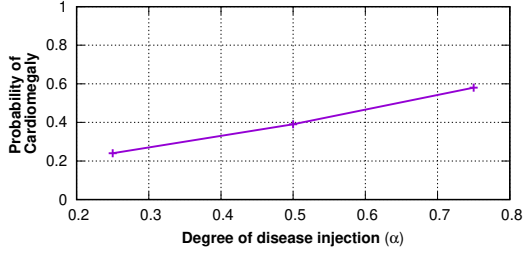Figure 17. Examples of failure cases for Proliferative DR.

Figure 18. Monotonic increase in probability of Cardiomegaly as assigned by evaluation classifier C_e^d when increasing degree of disease injection.

Appendix B.
User Study

The purpose of this session is to understand whether the provided chest X-ray images contain a specific disease condition. More specifically, for each question, we would like you to analyze the provided chest x-ray image, and determine whether it indicates the disease condition of Cardiomegaly. To respond, please choose one of the following options:
1) 'Disease' if it is indicative of Cardiomegaly
2) 'No Disease' if it shows no signs of Cardiomegaly
3) 'Other'. Choose this only if you are not confident to make decisions 1) or 2). If you choose this option, please provide a brief explanation for your uncertainty. Note that 'Other' does not mean some other disease.
Do not make any assumptions about the number of images that fall in each category. Your decision should be solely based on the content of the image.

- There are 30 questions in this session. Once all questions are answered, please click the "Submit" button to complete the session.
- This is one of 5 survey sessions for this task.
- There is no time limit for completing the task, but we would request you to finish each session in one sitting.
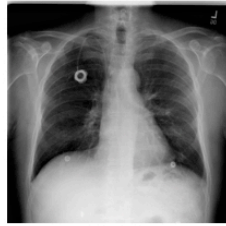
*

Image ID: 21

○ No Disease

○ Disease

○ Other:

Figure 19. Screenshot of User Study Interface for Part 1.

TABLE 10. FALSE POSITIVE RATE OF EVALUATORS WHEN (1) IDENTIFYING REAL X-RAYS WITH SYMPTOMS OF CARDIOMEGALY (FLEISS' $\kappa = 0.62$ ($p < 10^{-6}$)), AND (2) DISTINGUISHING BETWEEN REAL AND FAKE X-RAYS.

Evaluator	False Positive Rate	
	Study Part 1 (Real X-rays only)	Study Part 2
Evaluator 1	2.0%	9.0%
Evaluator 2	6.0%	7.0%
Evaluator 3	4.0%	8.0%
Average	4.0%	8.0%
Majority	3.0%	5.0%

TABLE 11. FALSE NEGATIVE RATE OF EVALUATORS WHEN (1) IDENTIFYING REAL X-RAYS WITH SYMPTOMS OF CARDIOMEGALY (FLEISS' $\kappa = 0.62$ ($p < 10^{-6}$)), AND (2) DISTINGUISHING BETWEEN REAL AND FAKE X-RAYS.

Evaluator	False Negative Rate	
	Study Part 1 (Real X-rays only)	Study Part 2
Evaluator 1	6.0%	43.0%
Evaluator 2	7.0%	45.0%
Evaluator 3	14.2%	37.0%
Average	9.1%	41.7%
Majority	6.0%	48.0%

For the X-ray shown below, choose one of the following options:
1) 'Real': A real X-ray is one produced by an X-ray imaging device, i.e., acquired through a medical imaging device. These are the X-rays analyzed by medical professionals.
2) 'Fake': These are X-rays generated using a computer algorithm, i.e., not acquired through a medical imaging device. Typically, computer generated images may lack 'realism', and may fail to capture important characteristics of real images.

You can use any basis for your judgement. However, do not make any assumptions about the number of images that fall in each class (real and fake). Your decision should be solely based on the content of the image.

Additionally, there is a text-box to provide an explanation for your decision. This is optional. However, we strongly encourage you to add a comment, especially if you choose 'Fake'.

*

Image ID: 1

○ Real

○ Fake

You can leave comments to explain your decision below. This is encouraged, especially if you select 'Fake'.

Your answer

Figure 20. Screenshot of User Study Interface for Part 2.

Appendix C.
Models and Datasets

C.1. Jekyll

Dataset preparation for Jekyll.
NIH Chest X-ray dataset. In order to effectively partition the dataset for both the attack and evaluation partitions, we decided upon a division that would allow construction of a reliable identity and disease classifier for the evaluation partition. To achieve this goal, we construct partitions on a patient level, rather than an image level. More specifically, we first pick 2,000 patients such that each of these patients holds 10 or more images. This ensures that we can build a reliable evaluation identity classifier. However, we also require an abundance of images for each disease, in order to build effective evaluation disease classifiers. We thus add another 4,805 patients to the evaluation partition, to allow for a large

TABLE 12. GENERATOR AND DISCRIMINATOR ARCHITECTURES.

Generator Architecture						
Layer Type	# of channels	Filter Size	Stride	Normalization	Activation	Output Shape
padding	3	-	-	-	-	(262,262,3)
conv2d	64	7×7	1	instance_norm	relu	(256,256,64)
conv2d	128	3×3	2	instance_norm	relu	(128,128,128)
conv2d	256	3×3	2	instance_norm	relu	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
residual block	256	-	-	-	-	(64,64,256)
deconv2d	128	3×3	2	instance_norm	relu	(128,128,128)
deconv2d	64	3×3	2	instance_norm	relu	(256,256,64)
padding	3	-	-	-	-	(262,262,3)
conv2d	3	7×7	1	instance_norm	tanh	(256,256,3))
Discriminator Architecture						
conv2d	64	4×4	2	instance_norm	relu	(128,128,64)
conv2d	128	4×4	2	instance_norm	relu	(64,64,128)
conv2d	256	4×4	2	instance_norm	relu	(32,32,256)
conv2d	512	4×4	1	instance_norm	relu	(32,32,512)
conv2d	1	4×4	1	-	-	(32,32,1))

TABLE 13. TRAINING CONFIGURATIONS OF *Jekyll* FOR DIFFERENT DISEASES.

Dataset	λ_{adv}	$\lambda_{disease}$	$\lambda_{identity}$	λ_{cycle}	Epochs	Batch size	Optimizer
Cardiomegaly	20	50	25	200	100	1	Adam
Severe DR	5	5	20	200	150	1	Adam
Proliferative DR	5	10	20	200	160	1	Adam

TABLE 14. DATASET STATISTICS (NUMBER OF IMAGES) AND TRAINING CONFIGURATIONS FOR ATTACK (C_a^d) AND EVALUATION (C_e^d) DISEASE CLASSIFIERS.

Dataset	C_a^d			C_e^d			Epochs	Batch Size	Optimizer
	Train	Validation	Test	Train	Validation	Test			
Cardiomegaly	1,600	100	999	1,600	100	1,177	50	5	Adam
Effusion	7,000	1,454	1,500	11,676	2,502	2,502	50	5	Adam
Severe DR	1,419	100	609	1,462	100	627	20	32	Adam
Proliferative DR	1,374	100	590	1,339	100	575	20	32	Adam

TABLE 15. DATASET STATISTICS (NUMBER OF IMAGES) AND TRAINING CONFIGURATIONS FOR ATTACK (C_a^i) AND EVALUATION (C_e^i) IDENTITY CLASSIFIERS. TRAINING CONFIGURATION FOR BOTH C_a^i AND C_e^i IDENTITY CLASSIFIERS ARE THE SAME FOR ANY GIVEN DATASET.

Dataset	C_a^i			C_e^i			Epochs	Batch Size	Optimizer
	Train	Validation	Test	Train	Validation	Test			
X-Ray	6,249	1,598	1,945	13,162	3,369	4,108	1000	16	Adam
Retinal Fundus	3,928	1,000	3,072	4,157	1,000	3,249	80	3	Adam

amount of images for each disease. This brings the evaluation partition to a total of 6,805 patients. All remaining 24,000 patients in the NIH Chest X-ray dataset are used to train *Jekyll*.

Retinal Fundus images. The retinal fundus image dataset only provides 2 images per patient; one image for the left eye, and one image for the right eye. This limitation simplifies that division of patients for the attack and evaluation partitions. More specifically, we divide the patients equally between the attack and evaluation partitions, with each receiving 44,351 patients.

Implementation details for alternative architectures. *StarGAN.* To fit StarGAN into the *Jekyll* framework, as discussed in section 5.1.5, we must use a disease classification loss, and a perceptual identity loss. The disease classification loss objective is determined using predictions from the auxiliary domain classifier. This domain classifier shares weights with the adversarial discriminator. This objective is optimized via a modified version of the binary-cross entropy loss function, called the focal loss [77]. Focal loss modifies cross-entropy to downweight the loss coming from easily classified

samples, and upweight that coming from difficult samples. We use hyperparameter values of $\alpha = 1$ and $\gamma = 2$. Here, $alpha$ is simply a balancing weight, and γ is the *focusing parameter*, which determines the rate at which easily classified samples are downweighted in the loss [77]. The perceptual identity loss objective is determined using an external attack identity classifier. The architecture of this external identity classifier (built separately in PyTorch for StarGAN) is identical to that of the default *Jekyll* attack identity classifier. More specifically, the identity loss is computed using output extracted from the third pooling layer (precedes the fourth dense block in the DenseNet-121 architecture). The model is trained for 200,000 iterations with a batch size of 1 and the Adam optimizer. Learning rate is set to 10^{-4}. We weight the loss objectives with weights $\lambda_{adversarial} = 20, \lambda_{disease} = 5, \lambda_{identity} = 0.1, \lambda_{gradient\_penalty} = 10, \lambda_{cycle} = 200$.

IPCGAN. To fit IPCGAN into the *Jekyll* framework, we must again introduce a disease classification loss, and a perceptual identity loss. IPCGAN already optimizes an age-loss, and we substitute the AlexNet age-classifier for an external disease classifier. Similarly, IPCGAN already optimizes a perceptual identity loss, and we substitute the identity classifier for an external identity classifier. Again, the identity loss is computed using output extracted from the third pooling layer (precedes the fourth dense block in the DenseNet-121 architecture). Finally, the generator input layer is changed to accept 256x256x3 input, instead of 128x128x3, as used in the original IPCGAN implementation. The model is trained for 200,000 iterations, with a batch size of 8 and the Adam optimizer. Learning rate is set to 10^{-4}. We weight the loss objectives with weights $\lambda_{adversarial} = 1, \lambda_{disease} = 1, \lambda_{identity} = 0.0005$.