Attack as the Best Defense: Nullifying Image-to-image Translation GANs via Limit-aware Adversarial Attack

Chin-Yuan Yeh[1,3], Hsi-Wen Chen[1], Hong-Han Shuai[2], De-Nian Yang[3], Ming-Syan Chen[1]
[1]National Taiwan University [2]National Yangming Jiaotong University [3]Academia Sinica
{d09942009, d09921004, mschen} @ntu.edu.tw hhshuai@g2.nctu.edu.tw dnyang@iis.sinica.edu.tw

Abstract

Due to the great success of image-to-image (Img2Img) translation GANs, many applications with ethics issues arise, e.g.*, DeepFake and DeepNude, presenting a challenging problem to prevent the misuse of these techniques. In this work, we tackle the problem by a new adversarial attack scheme, namely the* ***Nullifying Attack****, which cancels the image translation process and proposes a corresponding framework, the* ***Limit-Aware Self-Guiding Gradient Sliding Attack (LaS-GSA)*** *under a black-box setting. In other words, by processing the image with the proposed LaS-GSA before publishing, any image translation functions can be nullified, which prevents the images from malicious manipulations. First, we introduce the* ***limit-aware RGF*** *and the* ***gradient sliding mechanism*** *to estimate the gradient that adheres to the adversarial limit,* i.e.*, the pixel value limitations of the adversarial example. We theoretically prove that our model is able to avoid the error caused by the projection in both the direction and the length. Then, an effective* ***self-guiding prior*** *is extracted solely from the threat model and the target image to efficiently leverage the prior information and guide the gradient estimation process. Extensive experiments demonstrate that LaS-GSA requires fewer queries to nullify the image translation process with higher success rates than* 4 *state-of-the-art methods.*

1. Introduction

Recently, Generative Adversarial Networks (GANs) [13] have achieved impressive breakthroughs on various image-to-image translation (Img2Img) tasks, including inpainting [25] and style transfer [33]. These models learn the cross-domain mapping by ensuring that the style of translated images is close to the image style of the target domain while the semantics of the input image are still preserved, *e.g.*, the identity or the layout.

[1]Due to ethical reasons, the illustrative examples utilizes hair-changing model BLACK2BLOND, instead of DeepNude [31].

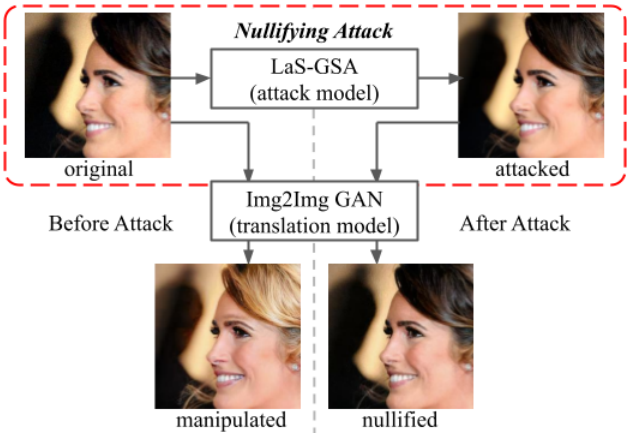


Figure 1: An illustration of nullifying attack against Img2Img GAN. The original portrait is initially manipulated by model BLACK2BLOND to impaint the portrait image with blond hair.[1] After the nullifying attack, LaS-GSA adds human imperceptible perturbation to the original image and generate an attacked image, which leads the Img2Img GAN to return the nullified image with black hair identical to the original image.

However, Img2Img GANs have also been misused to generate fake images, *i.e.*, DeepFake [16] and DeepNude [12]. For example, DeepNude excels in undressing full-body shots and producing realistic nude images. Facing the threat of these immoral algorithms, a simple way is to detect DeepFake contents [26, 31] after the fake images are released. However, even though those post-detection methods can catch the footprints of DeepFake, the manipulated images have already harmed each individual's reputation. Our idea is to defend personal privacy in the first place by nullifying the translation process of misused Img2Img GANs. We aim to attach human-imperceptible perturbations to input images, such that the attacked image can be refrained from being immorally manipulated (to produce obscene images with DeepFake). Thus, our goal is to conduct adversarial attacks against misused Img2Img GANs.

To develop an adversarial attack against misused Img2Img GANs, a simple approach is adopting the *Distorting Attack* [27, 10], which distorts the image translation process of the Img2Img GANs to generate a deteriorated image. However, it can lead to unpredictable results in this case. For example, if the distorting attack is applied to *e.g.*, DeepNude, the distorted regions may appear in the background, and naked images are still created after cropping [32]. Therefore, in this paper, we introduce a new attack, namely the *Nullifying Attack*, in a black-box setting.[2] Compared with the distorting attack, the nullifying attack is designed to cancel the translation process of misused Img2Img GANs and generate an output image nearly identical to the input one. Figure 1 illustrates the nullifying attack, where the targeted Img2Img GAN is nullified by the adversarial example created by our attacked method (detailed later).[1]

To facilitate nullifying attack in a black-box setting, one approach is to exploit surrogate models to approximate gradient [19, 24, 11], *i.e.*, the optimal modification to generate a successful adversarial example. However, preparing surrogate models for an Img2Img GAN requires additional computational resources, and the datasets need to be preprocessed and prepared for model training.[3] Moreover, creating another surrogate model with functions similar to the threat model is morally questionable when the threat models are unethical Img2Img GANs.

On the other hand, query-based attacks [6] estimate the gradient for modifying the image by querying the target model and conducting zeroth-order optimization. However, such attacks are inefficient because they usually require more than 10^6 queries to optimize the adjustment of each pixel for an RGB image. While acceleration schemes have been proposed for the adversarial attack against image classifiers [30, 1], the adversarial attack against Img2Img GANs is more challenging because it is required to alter the entire output image to a visually distinguishable degree, instead of simply changing a few labels in image classification [27].

To address the above challenge, we introduce *Limit-Aware Self-Guiding Gradient Sliding Attack (LaS-GSA)* to attack Img2Img GANs effectively. First, we prove that naively projecting the gradient, *i.e.*, clipping the gradient [7, 30] to achieve human-imperceptible modifications, has a detrimental effect on the correctness of the nullified process. Therefore, a *limit-aware strategy* is devised to avoid querying the gradient in the directions that violate the *adversarial limit*, *i.e.*, the pixel value limitations of an adversarial example to follow the imperceptible requirement. Then, a *gradient-sliding mechanism* is introduced to extend the modification along the boundary of the adversarial limit and avoid being trapped in the limit boundary, such that the nullifying attack can be achieved efficiently. Last, by investigating the semantic consistency of Img2Img GANs, we present the *self-guiding prior* that can be extracted from the targeted model directly and remove the cost of preparing surrogate models. At the same time, valuable information is still obtained by the prior to facilitate the nullifying attack in a black-box setting.

The contributions of this paper are as follows:

- We introduce a new adversarial attack on Img2Img GANs, namely the *Nullifying Attack*, and propose the LaS-GSA to cancel the translation process in a black-box setting.
- We investigate the detrimental effects of the projection for the adversarial limit and propose the *limit-aware RGF* and the *gradient sliding mechanism* to effectively mitigate the harm in the gradient estimation process.
- With the *self-guiding prior*, we provide an efficient scheme to extract prior information from Img2Img GANs, removing the need for surrogate models.
- Experimental results demonstrate the effectiveness and efficiency of LaS-GSA compared with 4 state-of-the-art methods on 3 Img2Img GANs.

2. Preliminary

2.1. Image-to-image translation GANs

The goal of image-to-image translations [2] is to learn a mapping $\mathbb{T}$ that translates an image x from an input domain X to a target domain Y, *i.e.*, $\mathbb{T}(x) = y \in Y \ \forall x \in X$. As Generative Adversarial Networks (GANs) [13] have been demonstrated to be effective in synthesizing realistic images, Img2Img GANs [15, 18] have been widely adopted to develop state-of-the-art image-to-image translation models. The objective of Img2Img GANs is as follows,

$$\min_G \max_D \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_x[\log(1 - D(x, G(x)))], \quad (1)$$

where the generator G learns to translate x into a realistic target domain sample, and the discriminator D learns to differentiate between a real y and a translated example $G(x)$. While the training is allowed to be conducted either in a supervised setting (*e.g.*, pix2pix [15]) or in an unsupervised setting [18], we first explore the latter due to its higher versatility. CycleGAN [33], an unsupervised Img2Img GAN, trains a pair of generator G to translate in both directions between the source and target domains. During inference time, we adopt the trained generator G on the specified direction as the targeted translation function $\mathbb{T}$. The nullifying

[2] Since the white-box attack requires the complete knowledge of the threat model, including the model architectures and weights, we focus on the black-box attack which is more practical in real-world applications (*e.g.*, Google Cloud Vision) [24, 9].

[3] For instance, training a CycleGAN model involves collecting thousands of relevant images and hundreds of epoch of training on a pair of models with 10^7 parameters [33].

attack is designed to create an adversarial image $x' \approx x_0$ such that $\mathbb{T}$ cannot translate x' to $y \in Y$, but returns the original input x_0 nearly unchanged after translation.

2.2. Projected gradient descent for adversarial attack

Given a neural network $f(x)$ and an input-output pair (x, y), the objective of an adversarial attack is to find an adversarial example x^* that 1) does not generate the expected output $f(x^*) \neq y$, 2) is a legitimate image, and 3) is within the norm-bounded region centering x with a small range $\epsilon \ll 1$ measured in the ℓ_p norm,[4] *i.e.*,

$$f(x^*) \neq y, \ s.t. \ x^* \in [0,1]^N \wedge \|x^* - x\|_\infty \leq \epsilon, \quad (2)$$

where N is the image dimension, $[0,1]^N$ is the N-orthotope, defined by the legitimate range of values for each pixel (*i.e.*, the *prefix limit*), and $\|x^* - x\|_\infty \leq \epsilon$ is the N-sphere centered at x with radius ϵ measured in the ℓ_∞-norm $\|\cdot\|$ defined according to the requirement for the perturbation to be human-imperceptible (*i.e.*, the *norm-bound limit*). We denote the union of the two limits as the adversarial limit $\Omega \equiv [0,1]^N \wedge \|x^* - x\|_\infty$ (illustrated in Figure 2(a)).

The adversarial example is generated by solving the constrained optimization problem

$$x^* = \arg\min_{x' \in \Omega} \mathcal{L}(x'), \quad (3)$$

where $\mathcal{L}$ is the adversarial loss representing the attack objective, *e.g.*, nullify the functionality of the Img2Img GAN and keep the input unchanged after translation.

To solve Eq. (3), many gradient-based methods [5, 20, 28] have been proposed, among which projected gradient descent (PGD) is proven best relying only on first order information [20]. PGD iteratively conducts gradient descent and projection to advance toward the optimal while remaining within the constrained regions. Specifically, let x_t^* and g_t denote the adversarial example and the gradient at the t^{th} iteration, respectively. The adversarial example at the $t+1^{\text{th}}$ iteration becomes

$$x_{t+1}^* = \Pi(x_t^* + \eta g_t), \ g_t = \frac{\nabla \mathcal{L}(x_t^*)}{\|\nabla \mathcal{L}(x_t^*)\|_2}, \quad (4)$$

where Π is the projection operation onto the adversarial limit Ω, *i.e.*, clipping the modification back to the adversarial limit [24].

2.3. Black-box setting and random gradient-free estimation

Since DeepFake models are generally concealed, nullifying attack naturally occurs in a black-box setting, which only allows one to acquire zeroth-order information, *i.e.*, the system output of a specific query. Therefore, to properly exploit gradient descent optimization, we perform zeroth-order estimations of the gradient by leveraging the Random Gradient-Free (RGF) estimation [22]. RGF randomly selects query vectors u_i from a unit sphere $\mathcal{U}$ to estimate a gradient $\hat{g}_t$ via

$$\hat{g}_t = \frac{1}{q} \sum_{i=1}^{q} \frac{L(x_t^* + \delta u_i) - L(x_t^* - \delta u_i)}{2\delta} u_i, u_i \in \mathcal{U}, \quad (5)$$

where δ is a small variance. In Eq. (5), the querying vectors are *flipped* towards the gradient by the multiplication of their own dot product with the gradient. Thus, by querying with radial symmetry, other directions orthogonal to the gradient will be balanced-out in the process to estimate the gradient for nullifying attack effectively.

3. Problem formulation

For Img2Img GANs, the adversarial attack objective is expressed naturally by shifting the output of the image translation process relative to an attack target y_{target}, with the corresponding adversarial loss L_{adv} defined as,

$$L_{adv}(x^*) = d(\mathbb{T}(x^*), y_{\text{target}}), \quad (6)$$

where d is the function of squared Euclidean distance, *i.e.*, $d(x, y) = (\|x - y\|_2)^2$. By minimizing the loss, the attack model is able to generate an adversarial example x^* that causes the translation function to returns output similar to the target image y_{target}. In the following, we formally introduce the nullifying attack.[5]

Definition 1. ***Nullifying attack.*** *The nullifying attack aims to nullify the image translation process such that the adversarial example x^* is mapped back to the original input x_0, according to the nullifying loss $L_{Null} = \|\mathbb{T}(x^*) - x_0\|_2^2$.*

A successful nullifying attack can be adopted as a watermark on personal images such that unethical Img2Img GANs (*e.g.*, DeepNude) cannot manipulate the image.[6]

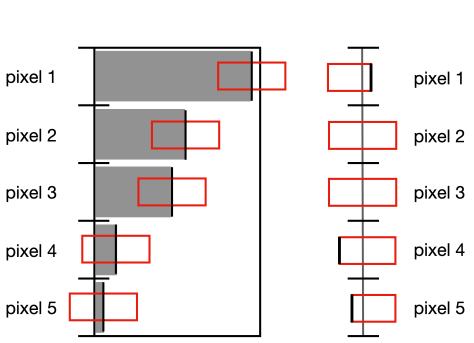
4. The LaS-GSA method

In the following, we introduce the *Limit-Aware Self-Guiding Gradient Sliding Attack (LaS-GSA)* scheme, a new black-box adversarial attack, to efficiently nullify the translation process of Img2Img GANs. First, the detrimental effects caused by the projection are investigated, leading

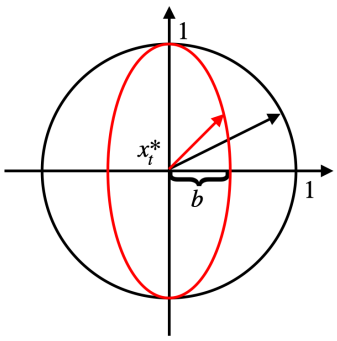
[4] $p = 2$ or ∞ is the common choice for adversarial attacks. In this paper, we adopt $p = \infty$ because it simplifies the projection to pixel-by-pixel numerical upper and lower bounds.

[5] Compared with attacking a classifier, which only alternates a single output label [30], it is more challenging to attack Img2Img GAN because the attack model is required to ensure the correctness of 10^6 pixels [33].

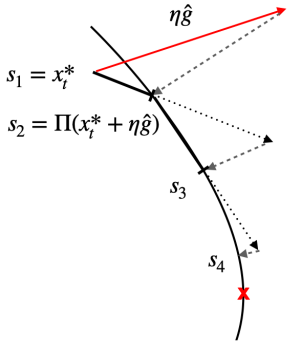
[6] We discuss another attack scheme, *Distorting Attack*, which forces the model generates the deteriorated output image in Appendix D.



(a) Left: grey bars denote pixel values, the outer black box denotes the *prefix limit* (*e.g.*, $[0, 1]$), and the red box denotes the *norm-bound limit* (in ℓ_∞). Right: the combined *adversarial limit* is depicted with centered pixel values.



(b) In RGF, random vectors are queried from the unit circle (black). In limit-aware RGF, the queried vector is shifted to the origin-centered ellipse (red).



(c) Starting from $s_1 = x_t^*$, PGD moves to s_2, whereas gradient sliding selects s_4, closer to the optimum (red x).

Figure 2: Illustrations of (a) adversarial limit, (b) limit-aware RGF, and (c) gradient sliding mechanism.

to the introduction of the *limit-aware RGF* and the *gradient sliding mechanism*, designed to alleviate the harmful effects. Then, we propose the *self-guiding prior* to fully exploit the threat model for prior information by deriving the approximate solution of the true gradient, removing the requirement for surrogate models or extra datasets [7]. Last, we present the attack procedure of the LaS-GSA method.

4.1. Limit-aware RGF

While the combination of RGF estimation and PGD optimization had been studied in previous black-box attack methods [30, 3, 7], they do not consider the detrimental effects of the projecting *i.e.*, clipping, the modification back to the adversarial limit. While the projection is necessary for keeping the adversarial example valid and indistinguishable from the original image, it not only deteriorates the efficiency of the gradient estimation process but also shortens the desired modification towards the estimated gradient, because the projection *pulls back* the out-of-bound gradient. Therefore, the adversarial example is modified towards an undesirable direction, which reduces the effectiveness of both the RGF estimation process and the gradient descent process in PGD. Therefore, we characterize the detrimental effects of projection in twofold: i) misdirection of the gradient, and ii) shortening of the optimization steps.

First, we prove that the projection would mislead the direction of the estimated gradient, harming the efficiency of the nullifying attack process.

Proposition 1. *(Proof in Appendix A.1.) The projection has a detrimental effect on the gradient estimation,* i.e., $g \cdot (\Pi(\hat{g}) - \hat{g}) \leq 0$.

To alleviate the detrimental effects of projection, we introduce the *limit-aware RGF* to query the vectors following the adversarial limit, *i.e.*,

$$\Pi(u_i) = u_i \, \forall u_i \rightarrow \Pi(\hat{g}) = \hat{g}. \tag{7}$$

By examining the convexity of the adversarial limit (detailed in Appendix A.3), the estimated gradient will not exceed the limit. Based on the observation, we *adjust* the unit N-sphere $\mathcal{U}$ in Eq. (5) to follow the adversarial limit by scaling the basis of $\mathcal{U}$ into a hyperellipsoid $\mathcal{P}$.[7]

Concretely, since adversarial limit Ω forms an N-orthotope [2, 24], we carefully transform the coordinate system to: 1) set the origin to the current adversarial example x_t^*, and 2) adopt every pixel as an independent basis to build an orthonormal basis of the $\mathbb{R}^N$ space. Thus, Ω becomes an axis-aligned hyperrectangle that includes the origin. Let Ω_i denote the corresponding range on the i^{th} axis of the N-orthotope Ω. To maintain radial symmetry, we define the scale vector b as a vector with the i^{th} element b_i indicating the adjustment range (to increase and decrease the pixel value) for i^{th} pixel, *i.e.*,

$$b_i = (\Omega_i^+, \Omega_i^-)/2, \;\; \Omega_i^+ \equiv \max(\Omega_i), \;\; \Omega_i^- \equiv -\min(\Omega_i), \tag{8}$$

$$\mathcal{P} = \{x \in \mathbb{R}^N : \sum_{i=1}^{N} \frac{x_i^2}{b_i^2} = 1\}. \tag{9}$$

As illustrated in Figure 2(b), we scale the unit $N-$sphere into the hyperellipsoid $\mathcal{P}$. Equipped with $\mathcal{P}$, the estimated gradient $\hat{g}_t$ can be formally written as follows,

$$\hat{g}_t = \frac{1}{q} \sum_{i=1}^{q} \frac{L_{adv}(x_t^* + \delta u_i) - L_{adv}(x_t^* - \delta u_i)}{2\delta} u_i, \;\; u_i \in \mathcal{P}. \tag{10}$$

[7] Recall that in Section 2.3, the query space of query vector $\mathcal{U}$ is required to exhibit radio symmetry.

By the convexity of the adversarial limit, $\hat{g}_t$ satisfies the adversarial limit and maintains radial symmetry by adjusting the range for increasing and decreasing the pixel value simultaneously. Moreover, by adding the scale vector, restricted pixels are effectively squeezed, and thus more adjustments can be facilitated for less restricted pixels.

4.2. Gradient sliding mechanism

In addition to the direction of the estimated gradient, we prove that the projection also shortens the gradient step.

Proposition 2. *(Proof in Appendix A.2.) The absolute length of the projection result is smaller than the original estimated gradient vector,* i.e., $\|\Pi(\hat{g})\|_2 \leq \|\hat{g}\|_2$.

Thus, we propose the *gradient sliding mechanism* to expand each projected gradient step into a series of sliding-steps $\{s_i\}_{i=1}^{M}$, where M is the number of steps. As illustrated in Figure 2(c), instead of being trapped by the adversarial limit, the sliding-steps circumvent along the limit boundary.[8] We carefully configure the steps such that the total length of these sliding-steps is approximately the original gradient step length before projection $l \equiv \|\eta\hat{g}\|_2$. At step $t+1$, the gradient sliding mechanism starts from the previous adversarial example x_t^* and the new adversarial example $\Pi(x_t^* + \eta\hat{g})$ and iteratively derive the next sliding-steps from the previous two sliding-steps, *i.e.*,

$$\begin{aligned} &s_1 = x_t^*,\ s_2 = \Pi(x_t^* + \eta\hat{g}), \\ &l_i = \max(0, l - \sum_{k=1}^{i} \|s_k - s_{k-1}\|_2), \\ &s_i = \Pi(s_{i-1} + l_i \cdot (s_{i-1} - s_{i-2})). \end{aligned} \tag{11}$$

Note that we still adopt projection on the sliding-steps to follow the adversarial limits (detailed in Appendix A.4). The sliding process terminates when the sum of trajectory length exceeds l. Since the sliding-step doesn't invoke new queries to the threat model, adopting the gradient sliding mechanism for the nullifying attack does not require additional queries compared with the conventional PGD [20].

4.3. Self-guiding prior

Although we have addressed the adversarial limit by querying from the limit-aware hyperellipsoid as well as performing the gradient sliding mechanism, nullifying attack is still difficult to be achieved without effective prior information due to the larger search space, *i.e.*, every possible modification of each pixel on the entire image. While several studies [7, 21] utilize a transfer-based prior that requires a surrogate model trained on extra datasets, it is computationally expensive to prepare a surrogate model. In contrast, by carefully investigating the nullifying process, Img2Img GANs can be exploited as a self-guide because of the semantic consistency of the translation process [33].

From Definition 1, the gradient of the nullifying attack at the t^{th} step can be derived as,

$$\nabla L_{Null}(x_t^*) = 2\mathbf{J}^T(\mathbb{T}(x_t^*) - x_0), \tag{12}$$

where the Jacobian matrix transposed $\mathbf{J}^T$ is multiplied to a *discrepancy vector*, *i.e.*, the difference between the current output $\mathbb{T}(x_t^*)$ and the input image x_0, the desired change for the adversarial output [4] (detailed in Appendix A.5).

Due to the semantic consistency of Img2Img GANs [33], perturbations to each input pixel mostly affect the same pixel in the output [2]. Thus, the Jacobian matrix $\mathbf{J}$ is sufficiently diagonal and it is promising to approximate $\mathbf{J}^T$ with $\mathbf{J}$.[9] Let a denote the discrepancy vector $\mathbb{T}(x_t^*) - x_0$. We estimate the gradient by right multiplying the discrepancy vector to the Jacobian matrix $\mathbf{J}a$. However, $\mathbf{J}a$ is simply the result of feeding the discrepancy vector into the Img2Img GANs (detailed in Appendix A.6). We thus arrived at a suitable self-guiding prior v,

$$v \equiv \mathbf{J}a \approx \frac{\|a\|_2(\mathbb{T}(x_t^* + \delta\hat{a}) - \mathbb{T}(x_t^*))}{\delta},\ \hat{a} = \frac{a}{\|a\|_2}. \tag{13}$$

With the above approximation, we significantly reduce the time complexity from the $O(N^2)$ to $O(1)$ for the Jacobian transposed $\mathbf{J}^T$ to find a self-guiding prior without exploiting additional surrogate models [7, 21], effectively boosting the nullifying attack process.

4.4. Optimization strategy

Equipped with the limit-aware RGF, the gradient sliding mechanism and the self-guiding prior, we present the final optimization strategy of *Limit-Aware Self-Guiding Gradient Sliding Attack (LaS-GSA)*. Our self-guiding prior is integrated into the RGF and the PGD framework by querying random vectors u_i biased towards the self-guiding prior v,

$$u_i = \sqrt{\lambda}\hat{v} + \sqrt{1-\lambda}\hat{t}_i,\ t_i = (\xi_i - (\hat{v}\cdot\xi_i)\hat{v}),\xi_i \in \mathcal{P}, \tag{14}$$

where $\hat{t}_i = \frac{t_i}{\|t_i\|_2}$, $\hat{v} \equiv \Pi(\frac{v}{\|v\|})$ is the projected prior, and $\lambda \in [0,1]$ controls the bias of the query u_i towards the prior $\hat{v}$.[10] Each query vector u_i is plugged into Eq. (5) to estimate the gradient and conduct the PGD process in Eq. (4). After each gradient step, we perform the sliding-step in Eq. (11). Utilizing the three techniques, LaS-GSA effectively and efficiently nullifies the targeted Img2Img GAN model. The pseudocode is presented in Algorithm 1.

[8] While the gradient step $\Pi(\eta\hat{g}_t)$ is compressed from the estimated gradient $\eta\hat{g}_t$ in Eq. (5), the sliding-steps s_i (Eq. (11)) expand the gradient step along the boundary to recover the full length of $\eta\hat{g}_t$.

[9] The diagonality of the Jacobian matrix $\mathbf{J}$ is evaluated in Appendix C.

[10] The optimal λ is explained and derived in Appendix B.

Algorithm 1 Limit-Aware Self-Guiding Gradient Sliding Attack (LaS-GSA)

Require: The translation model $\mathbb{T}$, input image x_0, projection operation Π, sampling variance δ, query number q, iteration number e, sliding-step number M, the learning rate η.

Ensure: The adversarial example x^*

$x^* \leftarrow x_0$
for $i = 1$ to e **do**
  $\hat{a} \leftarrow \Pi(\frac{a}{\|a\|}), a = \mathbb{T}(x^*) - x_0, \hat{g} \leftarrow 0,$
  $\hat{v} \leftarrow \Pi(\frac{v}{\|v\|}), v = \frac{1}{\delta}\left(\mathbb{T}(x^* + \delta \cdot \hat{a}) - \mathbb{T}(x^*)\right)$
  Find b according to Eq. (8)
  Estimate λ^* with $\mathbb{T}$, $\hat{v}$, q according to [7]
  for $j = 1$ to q **do**
    Uniform sample r_j from the unit N-sphere $\mathcal{U}$;
    $\xi_j = b \circ r_j$
    $t_j \leftarrow \xi_j - (\hat{v} \cdot \xi_j)\hat{v}$
    $u_j = \sqrt{\lambda^*}\hat{v} + \sqrt{1-\lambda^*}t_j$
    $\hat{g} \leftarrow \hat{g} + \frac{1}{\delta}\left((\mathbb{T}(x^* + \delta u_j) - x_0)^2 - a^2\right)$
  $x_{\text{prev}} \leftarrow x^*$, $x_{\text{curr}} \leftarrow \Pi(x^* + \eta \cdot \frac{1}{q}\hat{g})$,
  $l \leftarrow \|\eta \cdot \frac{1}{q}\hat{g}\|_2$, $l_{\text{slide}} \leftarrow 0$
  for $k = 1$ to M **do**
    $\xi \leftarrow \max(0, l - l_{\text{slide}})$
    if $\xi = 0$ **then**
      break
    $x_{\text{next}} \leftarrow \Pi(x_{\text{curr}} + \xi \cdot (x_{\text{prev}} - x_{\text{curr}}))$
    $x_{\text{prev}} \leftarrow x_{\text{curr}}$
    $x_{\text{curr}} \leftarrow x_{\text{next}}$
    $l_{\text{slide}} \leftarrow l_{\text{slide}} + \|x_{\text{curr}} - x_{\text{prev}}\|_2$
  $x^* \leftarrow x_{\text{curr}}$
return x^*

5. Experiments

We compare LaS-GSA with 4 state-of-the-art black-box adversarial attack schemes. All attack methods are implemented for 3 Img2Img GANs relevant to the manipulation of personal images: 2 trained on closed-up portraits and 1 trained on full-body shots. We first present the experiment setup. Then, we present quantitative and qualitative evaluations of the attack results and an ablation study.

5.1. Experimental setup

Threat Models. We adopt CycleGAN [18] as the default Img2Img GAN architecture for the following threat models: 1) BLACK2BLOND, which is trained on HQ-CelebA dataset [17] to translate people with black hair to blonde hair, 2) NONE2GLASSES, which adds glasses to portraits, also trained on HQ-CelebA dataset, and 3) BLUE2RED, which is trained on self-prepared datasets of clean images for people wearing blue and red shirts from Google Image Search for translating blue shirts to red shirts. Besides, we select 100 testing samples [29] that are i.i.d. to the training set of each threat model.[11]

Baselines. The proposed *LaS-GSA* is compared with 4 state-of-the-art methods. 1) *Bandit* [14] adopts the time-dependent prior vector to guide the sampling process. 2) *Square* [1] performs localized square-shaped updates at random positions. 3) *RGF* [22] randomly samples the query vectors from the unit N-sphere. 4) *Prior-RGF* [7] utilize the surrogate model to bias the query vectors in *RGF* towards the transfer-based prior vector estimated from the surrogate model.[12] The querying variance δ, norm-bound ϵ, and learning rate η are set to 0.001, 0.1, and 1, respectively. To provide transfer priors for the Prior-RGF method, surrogate models are prepared for each threat model with the *same* architectures and conditions.

Evaluations. To evaluate the results of different attack schemes, we present a task-oriented score, *i.e.*, the nullifying score s_{Null},

$$s_{\text{Null}}(x^*) = \left[1 - \frac{(\|\mathbb{T}(x^*) - x_0\|_2)^2}{\|y_0 - x_0\|_2^2}\right] \times 100, \qquad (15)$$

where the original translation distance $\|y_0 - x_0\|_2, y_0 = \mathbb{T}(x_0)$ acts as a normalization. Following [32], we consider adversarial examples x^* successful if $s_{\text{Null}}(x^*)$ is greater than the threshold 75.[13] The attack success rate (ASR) is defined as the percentage of the successful attack on test images in $100,000$ query budgets. The query count (Q) represents the average number of attempted queries (stopping upon passing the threshold) for each example.

5.2. Quantitative evaluations

Table 1 compares the proposed LaS-GSA against baseline methods in terms of the attack success rate (ASR) and the query count (Q) of the 100 testing images for each threat model. For all threat models, LaS-GSA outperforms all the other approaches in both ASR and Q. Remarkably, Bandit attack could not pass the threshold score within a $100,000$ query budget for some threat models. Compared to RGF, LaS-GSA also consistently achieves better performance in both ASR and Q by at least 10%, because LaS-GSA carefully examines the clipping effect and exploits self-guiding prior to attack the CycleGAN effectively. Even though Prior-RGF is equipped with a surrogate model, which has the identical CycleGAN structure trained on an i.i.d. testing dataset to estimate the prior, LaS-GSA still outperforms

[11] Additional qualitative results of nullifying attack on the 3 Img2Img GANs, *i.e.*, BLACK2BLOND, NONE2GLASSES and BLUE2RED and distorting attack on 3 models, *i.e.*, STR2SEG, FACADE2LABEL, and NIGHT2DAY, are presented in Appendix E.

[12] The surrogate models are trained with the same architecture and procedure on 100 i.i.d. samples of the original training set.

[13] The threshold is determined by 100 samples with 50 users [32].

Methods \ Models	BLACK2BLOND		NONE2GLASSES		BLUE2RED	
	ASR	Q/(s)	ASR	Q/(s)	ASR	Q/(s)
Bandit [14]	0%	(2)	10%	90,019	0%	(5)
Square [1]	23%	87,196	40%	60,194	25%	80,778
RGF [22]	71%	53,237	88%	58,049	23%	85,969
Prior-RGF [7]	69%	51,330	78%	81,580	23%	80,463
LaS-GSA	**85%**	**42,917**	**95%**	**40,298**	**40%**	**79,934**

Table 1: Quantitative results of the black-box attack against Img2Img GANs with a limit of 100,000 queries. We report the attack success rate (ASR) and the query count (Q) for all 100 test samples. If the attack fails in all 100 test samples, the average final score (s) is presented with parentheses.

Methods	ASR	Q
RGF	71%	53,237
GSA	76%	48,849
S-RGF	81%	49,743
S-GSA	84%	43,694
LaS-RGF	80%	51,871
LaS-GSA	**85%**	**42,917**

Table 2: Ablation test results for BLACK2BLOND with the attack success rate (ASR) and the query count (Q).

Figure 3: Comparing attack methods with adversarial results for model BLACK2BLOND.

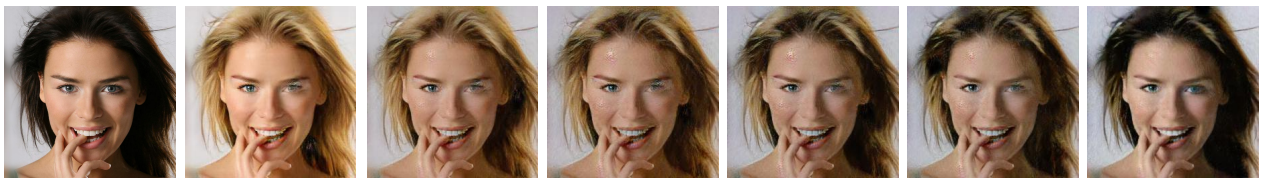


Figure 4: The attack process of LaS-GSA on BLACK2BLOND. From left to right: the original input image, expected Img2Img GAN output, and intermediate results of LaS-GSA attack, with the score (s_{Null})/query number shown below.

Prior-RGF by 2% to 17% regarding ASR. This is because the output space of GANs is much larger than image classifiers [32], and thus transferring the gradient across different models is much more challenging.

5.3. Ablation Study

Table 2 presents the ablation studies on BLACK2BLOND with RGF and four variants of our method, including 1) *GSA*: with only the gradient sliding mechanism, 2) *S-RGF*: with only the self-guiding prior, 3) *S-GSA*: with both the self-guiding prior and the gradient sliding mechanism, and 4) *LaS-RGF*: with both the limit-aware RGF and the self-guiding prior. First, variants equipped with the gradient sliding mechanism (*-GSA) consistently improve the performance by at least 3% regarding ASR compared with RGF. Besides, the self-guiding prior increases the overall

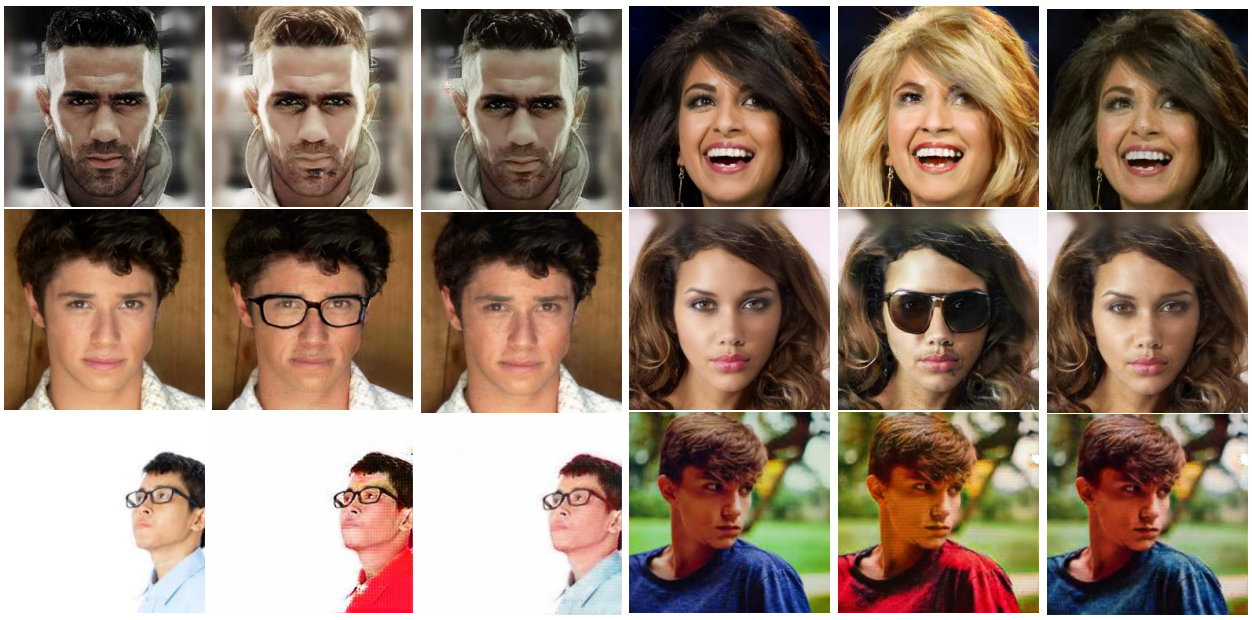


Figure 5: Qualitative LaS-GSA attack results against model BLACK2BLOND, NONE2GLASSES, and RED2BLUE from top to bottom, each presenting the input images, expected output of each model, and the final output after applying LaS-GSA.

ASR to 80%, and the limit-aware RGF improves ASR to 85%. Furthermore, the results in the query count (Q) follow a similar trend, in which LaS-GSA reduces by 20% of queries compared to RGF. Notice that LaS-GSA outperforms LaS-RGF by 17.2% regarding query efficiency since the gradient sliding mechanism carefully estimates the adversarial limit and prolongs the optimization steps along the constraint boundary, leading to better efficiency.

5.4. Qualitative evaluation

Figure 3 compares the visual quality of the attack results. Consistent with the quantitative results, Bandit and Square fail to alter the image output. Square blurs the image with the vertical stripes because their mechanism favors rectangle perturbations. For BLACK2BLOND, while other methods only slightly modify the hair color to *brown*, LaS-GSA is the only one that nullifies the translation process and returns a black hair image because it effectively utilizes the limit-aware gradient estimation and the gradient sliding mechanism to ensure the correctness of nullifying process in both direction and length. While LaS-GSA achieves similar results on NONE2GLASSES and BLUE2RED compared with RGF and Prior-RGF, as shown in Table 1, it requires fewer queries because the self-guiding prior can provide meaningful guide for the modification direction. Figure 4 visualizes the nullifying process of LaS-GSA on BLACK2BLOND, which recovers the hair color from blond to black. As query counts increase, the resulting output image shifts from blond hair back to black. We observe that a nullifying score $s_{\text{Null}} = 75$ is sufficient, with up to $30,000$ queries. Nonetheless, with $100,000$ queries, the nullifying attack can make the adversarial output much closer to the original input. Finally, Figure 5 further presents 2 additional samples for each threat model to demonstrate the generality of LaS-GSA. Our limit-aware strategy follows the adversarial limit and keep the adversarial perturbations imperceptible. With the nullifying attack scheme, LaS-GSA effectively creates adversarial examples that cause models BLACK2BLOND, NONE2GLASSES, and BLUE2RED to generate output images that are almost identical to the original input images, canceling the respective functionality. More qualitative results are presented in Appendix E.

6. Conclusion

In this work, we introduce a new adversarial attack on Img2Img GANs in a black-box setting, namely *Nullifying Attack*, to defend against malicious applications (*e.g.*, DeepFake). We propose the *Limit-Aware Self-Guiding Gradient Sliding Attack (LaS-GSA)* method, which incorporates the *limit-aware RGF*, the *gradient sliding mechanism*, and the *self-guiding prior* to cancel the image translation process of Img2Img GANs. Experimental results demonstrate the effectiveness and efficiency of our proposed method in 3 different translation processes. Future work includes reducing the vulnerability of Img2Img GANs against adversarial attacks for safety-critical applications.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020. 2, 6, 7

[2] Dina Bashkirova, Ben Usman, and Kate Saenko. Adversarial self-defense for cycle-consistent gans. In *Advances in Neural Information Processing Systems*, pages 635–645, 2019. 2, 4, 5

[3] Thomas Brunner, Frederik Diehl, Michael Truong Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4958–4966, 2019. 4

[4] Samuel R Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004. 5

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 3, 14

[6] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017. 2

[7] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. In *Advances in Neural Information Processing Systems*, pages 10932–10942, 2019. 2, 4, 5, 6, 7, 13, 14

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 16

[9] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *International conference on machine learning*, pages 1115–1124. PMLR, 2018. 2

[10] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 321–338, 2019. 2

[11] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 2

[12] github/lwlodo. Official deepnude algorithm source code, Jul 2019. 1

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2

[14] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2018. 6, 7

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[16] Mousa Tayseer Jafar, Mohammad Ababneh, Mohammad Al-Zoube, and Ammar Elhassan. Forensics and analysis of deepfake videos. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pages 053–058. IEEE, 2020. 1

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 6

[18] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 2, 6

[19] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *Proceedings of 5th International Conference on Learning Representations*, 2017. 2

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 3, 5

[21] Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *International Conference on Machine Learning (ICML)*, 2019. 5

[22] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017. 3, 6, 7

[23] Lukáš Neumann, Michelle Karg, Shanshan Zhang, Christian Scharfenberger, Eric Piegert, Sarah Mistr, Olga Prokofyeva, Robert Thiel, Andrea Vedaldi, Andrew Zisserman, et al. Nightowls: A pedestrians at night dataset. In *Asian Conference on Computer Vision*, pages 691–705. Springer, 2018. 16

[24] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017. 2, 3, 4

[25] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 1

[26] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *CVPR*, pages 1–11, 2019. 1

[27] Nataniel Ruiz, Sarah Adel Bargal, and Stan Sclaroff. Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems. In *ECCV*, pages 236–251. Springer, 2020. 2

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. 3

[29] Rohan Taori, Amog Kamsetty, Brenton Chu, and Nikita Vemuri. Targeted adversarial examples for black box audio systems. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 15–20. IEEE, 2019. 6

[30] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019. 2, 3, 4, 13, 14

[31] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, pages 8695–8704, 2020. 1

[32] Chin-Yuan Yeh, Hsi-Wen Chen, Shang-Lun Tsai, and Sheng-De Wang. Disrupting image-translation-based deepfake algorithms with adversarial attacks. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, pages 53–62, 2020. 2, 6, 7, 15

[33] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 1, 2, 3, 5, 15, 16

A. Proofs

A.1. Proof of Proposition 1

Proposition 1 *The projection has a detrimental effect on the gradient estimation, i.e.,* $g \cdot (\Pi(\hat{g}) - \hat{g}) \leq 0$.

Proof. We first prove that the inner product between the true gradient g and the RGF estimated gradient $\hat{g}$ is non-detrimental, *i.e.*, $g \cdot \hat{g} \geq 0$. According to Eq. (5), the estimated gradient is $\hat{g}_t = \frac{1}{q}\sum_{i=1}^{q} \frac{\partial L(x_t^*)}{\partial u_i} u_i$, where u_i is drawn from a radial symmetrical distribution $\mathcal{U}$, *e.g.*, the unit N-sphere. The partial derivative of $L(\cdot)$ at point x, with regard to u can be formally written as

$$\frac{\partial L(x)}{\partial u} = \frac{d}{d\zeta} L(x + \zeta u)|_{\zeta=0}, \tag{A.1}$$

where ζ is a dummy variable. Eq. (A.1) can be further expressed by the multivariable chain rule,

$$\begin{aligned} \frac{d}{d\zeta} L(x + \zeta u) &= \sum_j \frac{\partial L(x+\zeta u)}{\partial (x+\zeta u)_j} \frac{\partial (x+\zeta u)_j}{\partial \zeta} \\ &= \sum_j \frac{\partial L(x+\zeta u)}{\partial e_j} u_j, \end{aligned} \tag{A.2}$$

where e_j is the unit vector along the j^{th} basis. Since the dummy variable ζ is set to 0, according to Eq. (A.2), we find

$$\begin{aligned} \frac{\partial L(x)}{\partial u} &= \sum_j \frac{\partial L(x)}{\partial e_j} u_j = \left(\sum_j \frac{\partial L(x)}{\partial e_j} e_j \right) \cdot u \\ &\equiv \nabla L(x) \cdot u. \end{aligned} \tag{A.3}$$

Therefore, $g \cdot \hat{g}_t$ can be rewritten as,

$$\begin{aligned} g \cdot \hat{g}_t &= \nabla L(x) \cdot \left(\sum_{i=1}^{q} \frac{\partial L(x)}{\partial u_i} u_i \right) \\ &= \nabla L(x) \cdot \left(\sum_{i=1}^{q} (\nabla L(x) \cdot u_i) u_i \right) \\ &= \sum_{i=1}^{q} (\nabla L(x) \cdot u_i)^2 \geq 0. \end{aligned} \tag{A.4}$$

Next, we prove $\hat{g} \cdot (\Pi(\hat{g}) - \hat{g}) \leq 0$ by decomposing the projection function Π into the *prefix limit* projection function Π_Ω and the *norm-bound limit* projection function $\Pi_{\mathcal{B}(x,\epsilon)}$, *i.e.*,

$$\Pi(g) = \Pi_{\mathcal{B}(x,\epsilon)} \circ \Pi_\Omega(g). \tag{A.5}$$

For the prefix limit, since x^* is the optimal solution according to Eq. (2), $\|x^* - x\| \leq \epsilon$ holds. Therefore, if $\Pi_\Omega(\hat{g}) \neq \hat{g}$, we have

$$\|x^* - x + \hat{g}\| \geq \epsilon \geq \|x^* - x + \Pi_\Omega(\hat{g})\|, \tag{A.6}$$

and $\hat{g} \cdot (\Pi_\Omega(\hat{g}) - \hat{g})$ becomes

$$\begin{aligned} &(x^* - x + \hat{g}) \cdot ((x^* - x + \Pi_\Omega(\hat{g})) - (x^* - x + \hat{g})) \\ &- (x^* - x) \cdot (\Pi(\hat{g}) - \hat{g}). \end{aligned} \tag{A.7}$$

The first term in Eq. (A.7) is smaller or equal to zero by the following simplification.

$$\begin{aligned} &(x^* - x + \hat{g}) \cdot ((x^* - x + \Pi_\Omega(\hat{g})) - (x^* - x + \hat{g})) \\ &= (x^* - x + \hat{g}) \cdot (x^* - x + \Pi_\Omega(\hat{g})) - (x^* - x + \hat{g})^2 \\ &\leq \|x^* - x + \hat{g}\| \|x^* - x + \Pi_\Omega(\hat{g})\| - \|x^* - x + \hat{g}\|^2 \\ &\leq 0. \end{aligned} \tag{A.8}$$

Besides, the absolute value of the second term, *i.e.*, $(x^* - x) \cdot (\Pi(\hat{g}) - \hat{g})$, is smaller than the first term because $\|x^* - x\| \leq \epsilon \leq \|x^* - x + \hat{g}\|$, and thus $\hat{g} \cdot (\Pi_\Omega(\hat{g}) - g) \leq 0$. Note that if $\Pi_\Omega(\hat{g}) = \hat{g}$, then $\hat{g} \cdot (\Pi_\Omega(\hat{g}) - \hat{g}) = 0$. Therefore, $\hat{g} \cdot (\Pi_\Omega(\hat{g}) - \hat{g}) \leq 0$.

Afterwards, we prove that the norm-bound limit projection has a detrimental effect on the gradient estimation. Without loss of generality, we assume that $p = 2$.[14] Each pixel of the adversarial example x_i^* has an upper limit b_i^u and a lower limit b_i^l, representing the valid range of adjustment, *i.e.*, $(x_i^*)_{t+1} - x_i \in [b_i^l, b_i^u]$.[15] Then, $\Pi_{\mathcal{B}(x,\epsilon)}(\hat{g})_i$ can be expressed as $\min(\max(\hat{g}_i, b_i^l), b_i^u)$, *i.e.*,

$$\begin{aligned} \Pi_{\mathcal{B}(x,\epsilon)}(\hat{g})_i &= \min(\max(\hat{g}_i, b_i^l), b_i^u) \\ &= \begin{cases} b_i^u, & \hat{g}_i > b_i^l \text{ and } \hat{g}_i > b_i^u \\ \hat{g}_i, & \hat{g}_i \geq b_i^l \text{ and } \hat{g}_i \leq b_i^u \\ b_i^l, & \hat{g}_i < b_i^l \end{cases} \\ &= \begin{cases} b_i^u, & \hat{g}_i > b_i^u \\ \hat{g}_i, & b_i^l \leq \hat{g}_i \leq b_i^u \\ b_i^l, & \hat{g}_i < b_i^l \end{cases}. \end{aligned} \tag{A.9}$$

Therefore, $\hat{g} \cdot (\Pi(\hat{g}) - \hat{g})$ can be written as,

$$\begin{aligned} \hat{g} \cdot (\Pi_{\mathcal{B}(x,\epsilon)}(\hat{g}) - \hat{g}) &= \sum_i \hat{g}_i \cdot (\Pi_{\mathcal{B}(x,\epsilon)}(\hat{g}) - \hat{g})_i \\ &= \begin{cases} \hat{g}_i \cdot (b_i^u - \hat{g}_i), & \hat{g}_i > b_i^u \geq 0 \\ 0, & b_i^l \leq \hat{g}_i \leq b_i^u \\ \hat{g}_i \cdot (b_i^l - \hat{g}_i), & \hat{g}_i < b_i^l \leq 0 \end{cases}. \end{aligned} \tag{A.10}$$

Since each element of $\hat{g} \cdot (\Pi_{\mathcal{B}(x,\epsilon)}(\hat{g}) - \hat{g})$ is non-positive, $\hat{g} \cdot (\Pi(\hat{g}) - \hat{g}) \leq 0$. Moreover, as two projection operations Π_Ω and $\Pi_{\mathcal{B}(x,\epsilon)}$ are monotonic, the projection function Π that combines the two operations still satisfies the property, *i.e.*, $\hat{g} \cdot (\Pi(\hat{g}) - \hat{g}) \leq 0$. Finally, since $g \cdot \hat{g} \geq 0$, the inequality becomes

$$g \cdot (\Pi(\hat{g}) - \hat{g}) \leq (g \cdot \hat{g})(\hat{g} \cdot (\Pi(\hat{g}) - \hat{g})) \leq 0. \tag{A.11}$$

The proposition follows. □

[14] If $p = \infty$, the proof is the same as that of the prefix limit.

[15] It is worth noting that $b_i^l \leq 0 \leq b_i^u$, otherwise $(x_i^*)_t$ does not exist.

A.2. Proof of Proposition 2

Proposition 2 *The length of the projection operation result is smaller than the original estimated gradient vector, i.e.,* $\|\Pi(\hat{g})\|_2 \leq \|\hat{g}\|_2$.

Proof. Recall that under ℓ_∞ norm, the adversarial limit can be regarded as an N-orthotope containing the origin, with the limit in each basis defined as Eq. (8). Intuitively, gradient vector will reduce in length after being *trimmed off* at the limit. For ℓ_2 norm, the adversarial limit is the union of a ℓ_2 N-sphere and an N-orthotope. If the projected gradient is on the ℓ_2 N-sphere, the gradient's length (prior to projection) is larger or equal to the radius of the N-sphere, and thus $\|\Pi(\hat{g})\|_2 \leq \|\hat{g}\|_2$. □

A.3. Proof of Eq. (7)

Eq. (7) $\Pi(u_i) = u_i \, \forall u_i \rightarrow \Pi(\hat{g}) = \hat{g}$.

Proof. Recall that the estimated gradient is the weighted sum of the query vectors. The space within the adversarial limit is convex since it is the union of convex spaces (N-orthotope Ω and hypersphere $\mathcal{B}_p(x, \epsilon)$). Thus, by convexity the estimated gradient using query vectors within the adversarial limit will remain in the adversarial limit. □

A.4. Proof of ending statement in Section 4.2

Description *The gradient sliding process follows the adversarial limit boundary.*

Proof. According to Eq. (11), starting from s_2, sliding steps are the results of projection operations, and we have $s_i = \Pi(n) = \Pi(\Pi(n)) = \Pi(s_i)$. Therefore, each s_i is within the adversarial limit for $i \geq 2$. □

A.5. Proof of (12)

Description of Eq. (12) *Given the nullifying loss* $L_{Null} = -(\|\mathbb{T}(x^*) - x\|_2)^2$, *the gradient is,*

$$\nabla L_{\text{Null}}(x_t^*) = -2\mathbf{J}^T(\mathbb{T}(x_t^*) - x_0).$$

Proof. We expand L_{Null} for each pixel component and simplify the variables with $x = x_t^*$ and $y = x_0$ as

$$L_{\text{Null}} = -(\|\mathbb{T}(x) - y\|_2)^2 = -\sum_i^N (\mathbb{T}(x)_i - y_i)^2. \quad \text{(A.12)}$$

By the definition of the gradient operator,

$$\begin{aligned}\nabla L_{\text{Null}}(x) &= -\sum_j^N \frac{\partial}{\partial e_j}\left(\sum_i^N (\mathbb{T}(x)_i - y_i)^2\right)\hat{e_j} \\ &= -2\sum_{i,j}^N (\mathbb{T}(x)_i - y_i)\frac{\partial \mathbb{T}(x)_i}{\partial e_j}\hat{e_j}. \quad \text{(A.13)}\end{aligned}$$

However, the element $\mathbf{J}_{ij}$ of the Jacobian matrix is $\frac{\partial \mathbb{T}(x)_i}{\partial e_j}$. Therefore, the j^{th} element of Eq. (A.13) can be regarded as the result of the element-wise multiplication of the j^{th} column vector of $\mathbf{J}$ and the vector $\mathbb{T}(x) - y$. By transposing $\mathbf{J}$, column vectors are transposed into row vectors. Therefore, the multiplication along with $\hat{e_j}$, summed over j, becomes the right-multiplication of the column vector $\mathbb{T}(x) - y$ to the transposed Jacobian matrix $\mathbf{J}^T$. Thus,

$$\sum_{i,j}^N (\mathbb{T}(x)_i - y_i)\frac{\partial \mathbb{T}(x)_i}{\partial e_j}\hat{e_j} \equiv \mathbf{J}^T(\mathbb{T}(x) - y). \quad \text{(A.14)}$$

Eq. (12) is proved. □

A.6. Proof of Eq. (13)

Description of Eq. (13) *Given the Img2Img translation model* $\mathbb{T}$ *and the discrepancy vector* a, *the self-guiding prior vector* $v \equiv \mathbf{J}a$ *can be approximated as*

$$v \equiv \mathbf{J}a \approx \frac{\|a\|_2(\mathbb{T}(x_t^* + \delta\hat{a}) - \mathbb{T}(x_t^*)}{\delta}, \; \hat{a} = \frac{a}{\|a\|_2}.$$

Proof. By the definition of the Jacobian matrix,

$$\begin{aligned}v \equiv \mathbf{J}a &= \frac{\partial \mathbb{T}_j}{\partial e_i}a_i = \frac{d}{d\zeta}\mathbb{T}(x + \zeta e_i)_j|_{\zeta=0} \cdot a_i \\ &= \lim_{\delta\to 0}\frac{1}{\delta}\left(\mathbb{T}(x_t^* + (\zeta+\delta)e_i)_j|_{\zeta=0} - \mathbb{T}(x + \zeta e_i)_j|_{\zeta=0}\right)a_i, \end{aligned} \quad \text{(A.15)}$$

where e_i represents the i^{th} basis, ζ is a dummy variable, and δ is the infinitesimal value for the derivation. We first expand Eq. (A.15) into a Taylor series, *i.e.*,

$$\begin{aligned}v = \lim_{\delta\to 0}\frac{1}{\delta}\Bigg(&\mathbb{T}(x_t^* + \zeta e_i)_j|_{\zeta=0} + \frac{\partial \mathbb{T}(x_t^* + \zeta e_i)_j|_{\zeta=0}}{\partial e_i}\delta e_i \\ &+ O(\delta^2) - \mathbb{T}(x_t^* + \zeta e_i)_j|_{\zeta=0}\Bigg)a_i, \end{aligned} \quad \text{(A.16)}$$

where $O(\delta^2)$ denotes the higher-order terms. Without loss of generality, we approximate v by dropping the higher-order term $O(\delta^2)$. Since a_i is a scalar which can be placed into the numerical limit, we rewrite Eq. (A.16) as,

$$\begin{aligned}v &\approx \lim_{\delta\to 0}\frac{1}{\delta}\left(\frac{\partial \mathbb{T}(x_t^* + \zeta e_i)_j|_{\zeta=0}}{\partial e_i}\delta e_i\right)a_i \\ &= \lim_{\delta\to 0}\frac{1}{\delta}\left(\frac{\partial \mathbb{T}(x_t^* + \zeta e_i)_j|_{\zeta=0}}{\partial e_i}\delta\hat{a}\right)\|a\|_2 \\ &= \lim_{\delta\to 0}\frac{1}{\delta}\left(\mathbb{T}(x_t^* + \delta\hat{a})_j - \mathbb{T}(x_t^*)_j\right)\|a\|_2. \end{aligned} \quad \text{(A.17)}$$

By replacing the limit with a small δ, the approximation of Eq. (13) holds. □

B. Implementation detail of LaS-GSA method

In this section, we detail the prior guiding query framework and the selection of the optimal λ. Then, we extend the framework to support limit aware RGF. The radial symmetry property is also presented.

B.1. Prior guiding query and the optimal λ

Following [7], the random query vector u_i in Eq. (5) is biased towards the prior vector v by a hyperparameter $\lambda \in (1, 0)$, which can be expressed by

$$u_i = \sqrt{\lambda}\hat{v} + \sqrt{1-\lambda}\hat{t_i},\ t_i = (\rho_i - (\hat{v}\cdot\rho_i)\hat{v})\,,\ \rho_i \in \mathcal{U}, \quad \text{(B.1)}$$

where $\hat{v}$ and $\hat{t}_i$ represent the normalization of the vectors v and t_i, respectively. Note that the above equation can be regarded as projecting the random unit vector onto a *cone* revolving the prior vector v, since $\rho_i \in \mathcal{U}$.

The optimal λ minimizes the difference between the estimated and the true gradient [30], *i.e.*,

$$\lambda^* = \arg\min_{\lambda}\ \mathbb{E}[\|\nabla L(x) - \hat{g}\|_2^2]. \quad \text{(B.2)}$$

By the Pythagorean theorem, we rewrite Eq. (B.2) by minimizing the vector component of the true gradient *orthogonal* to the expected estimated gradient $\mathbb{E}[\hat{g}]$),

$$\lambda^* = \arg\min_{\lambda}\ \left(\|\nabla L(x)\|^2 - \left(\frac{\nabla L(x)\cdot\mathbb{E}[\hat{g}]}{\|\mathbb{E}[\hat{g}]\|_2}\right)^2\right), \quad \text{(B.3)}$$

where the second term is the vector component of the real gradient *parallel* to $\mathbb{E}[\hat{g}]$. As the estimated gradient $\hat{g}$ is the weighted average of the query vector u_i, $\hat{g}$ is expressed as,

$$\hat{g} = \frac{1}{q}\sum_i (\nabla L(x)\cdot u_i)u_i \equiv \frac{1}{q}\sum_i \hat{g}_i. \quad \text{(B.4)}$$

$$\mathbb{E}[\hat{g}] = \mathbb{E}[\hat{g}_i] = \mathbb{E}[u_i u_i^T]\nabla L(x), \quad \text{(B.5)}$$

where u_i^T denotes the transpose of the query vector u_i. By replacing u_i in Eq. (B.5) with Eq. (B.1),

$$\mathbb{E}[u_i u_i^T] = \mathbb{E}[\lambda\hat{v}\hat{v}^T + (1-\lambda)\hat{t_i}\hat{t_i}^T],\ t_i = (\xi_i - (\hat{v}\cdot\xi_i)\hat{v})\,. \quad \text{(B.6)}$$

Since t_i is orthogonal to v, $\hat{t_i}$ is a unit vector in both the original $\mathbb{R}^N$ vector space and the $\mathbb{R}^{N-1}$ vector space orthogonal to v. Following [7], we decompose $\hat{t_i}$ as $\sum_{j=1}^{N-1} a_j e_j$, where e_j denotes a vector basis orthogonal to $\hat{v}$. With $\sum e_j e_j^T = \mathbf{I} - \hat{v}\hat{v}^T$ and $\sum {a_j}^2 = 1$, $\mathbb{E}[\hat{t_i}\hat{t_i}^T]$ becomes

$$\begin{aligned}\mathbb{E}[\hat{t_i}\hat{t_i}^T] &= \mathbb{E}\left[\left(\sum_{j=1}^{N-1} a_j e_j\right)\left(\sum_{k=1}^{N-1} a_k e_k^T\right)\right] \\ &= \mathbb{E}[{a_j}^2]\sum e_j e_j^T = \frac{1}{N-1}(\mathbf{I} - \hat{v}\hat{v}^T).\end{aligned} \quad \text{(B.7)}$$

Besides, we also expand $\mathbb{E}[\|\hat{g}\|_2^2]$ by

$$\begin{aligned}\mathbb{E}[\|\hat{g}\|_2^2] &= \mathbb{E}[\|\hat{g} - \mathbb{E}[\hat{g}]\|_2^2] + \|\mathbb{E}[\hat{g}]\|_2^2 \\ &= \frac{1}{q}\mathbb{E}[\|\hat{g}_i - \mathbb{E}[\hat{g}_i]\|_2^2] + \|\mathbb{E}[\hat{g}_i]\|_2^2 \\ &= \frac{1}{q}\mathbb{E}[\|\hat{g}_i\|] + (1-\frac{1}{q})\|\mathbb{E}[\hat{g}_i]\|_2^2.\end{aligned} \quad \text{(B.8)}$$

After inserting Eq. (B.5) and Eq. (B.8) into Eq. (B.3),

$$\begin{aligned}\lambda^* &= \arg\min_{\lambda} {\|\nabla L(x)\|_2}^2\left(1 - \frac{\text{NUM}}{\text{DENOM}}\right), \\ \text{NUM} &= (\lambda\alpha^2 + \frac{1-\lambda}{N-1}(1-\alpha^2))^2, \\ \text{DENOM} &= (1-\frac{1}{q})(\lambda^2\alpha^2 + (\frac{1-\lambda}{N-1})^2(1-\alpha^2)) \\ &\quad + \frac{1}{q}(\lambda\alpha^2 + \frac{1-\lambda}{N-1}(1-\alpha^2)),\end{aligned} \quad \text{(B.9)}$$

where α denotes the cosine similarity between the estimated gradient and the true gradient, *i.e.*, $\alpha = \frac{v\cdot\nabla L(x)}{\|v\|_2\|\nabla L(x)\|_2}$. To find the maximum of the second term, we set its derivative with regard to λ as 0 and obtain λ^* as follows.

$$\lambda^* = \begin{cases} 0 & \text{if } \alpha^2 \le \dfrac{1}{N+2q-2} \\ 1 & \text{if } \alpha^2 \ge \dfrac{2q-1}{N+2q-2} \\ \dfrac{(1-\alpha^2)(\alpha^2(N+2q-2)-1)}{2\alpha^2 Nq - \alpha^4 N(N+2q-2) - 1} & \text{otherwise.} \end{cases} \quad \text{(B.10)}$$

Since the cosine similarity between u_i and v is regulated by $\sqrt{\lambda}$, if $\alpha \approx 1$, the estimated gradient $\hat{g}$ is similar to the true gradient and $\lambda \approx 1$; otherwise, if $\alpha \approx 0$, $\lambda \approx 0$, and the problem is reduced to estimating the gradient $\nabla L(x)$ in the $N-1$ dimension excluding the dimension along v.

Finally, since the inner product of a vector and the gradient can be approximated into the directional derivative,

$$\alpha = \frac{v\cdot\nabla L(x)}{\|v\|\|\nabla L(x)\|} = \frac{1}{\|\nabla L(x)\|}\frac{\partial L(x)}{\partial\hat{v}}. \quad \text{(B.11)}$$

On the other hand, the squared cosine similarity of two random vectors is expected to be $\frac{1}{N}$. Thus, by estimating the inner product squared and extracting out the gradient length term, the gradient length $\|\nabla L(x)\|$ can be calculated as,

$$\begin{aligned}\frac{1}{S}\sum_{i=1}^{S}\left(\frac{\partial L(x)}{\partial\rho_i}\right)^2 &= \|\nabla L(x)\|^2\frac{1}{S}\sum_{i=1}^{S}\left(\frac{\nabla L(x)}{\|\nabla L(x)\|}\cdot\rho_i\right)^2 \\ &\approx \frac{\|\nabla L(x)\|^2}{N},\ \rho_i \in \mathcal{U},\end{aligned} \quad \text{(B.12)}$$

where ρ_i is queried from the unit N-sphere $\mathcal{U}$, and S determines the number of vectors to query. Furthermore, the partial derivative in Eq. (B.11) can be estimated as

$$\frac{\partial L(x)}{\partial\hat{v}} \approx \frac{1}{\delta}(L(x+\delta\hat{v}) - L(x)),\ \delta \ll 1. \quad \text{(B.13)}$$

Thus, we can adopt the cosine similarity α to estimate the optimal λ^* in Eq. (B.10).

B.2. Limit-aware projection to the prior-guiding query

Appendix B.1 presents an effective method to exploit the information from a prior vector. Here, we extend the biased querying strategy to support the limit-aware RGF. First, Eq. (B.1) is modified by replacing the prior $\hat{v}$ with the projected prior $\Pi(\hat{v})$ and replacing the query vector $r_i \in \mathcal{U}$ with ξ_i in the hyperellipsoid $\mathcal{P}$. Thus, according to Eq. (14), we still obtain the optimal cone as described in the N-sphere case. Besides, $\mathcal{P}$ is scaled symmetrically to fit in the adversarial limit from the unit N-sphere $\mathcal{U}$. Thus, the new probability distribution on the cone is still symmetric and satisfies the radial symmetry requirement (detailed in Appendix B.3).

The new query vectors are given as

$$u_i = \sqrt{\lambda}\hat{v} + \sqrt{1-\lambda}\hat{t}_i,\ t_i = (\xi_i - (\hat{v}\cdot\xi_i)\hat{v}), \xi_i \in \mathcal{P}, \quad \text{(B.14)}$$

where $\hat{t}_i = \frac{t_i}{\|t_i\|_2}$, $\hat{v} \equiv \Pi(\frac{v}{\|v\|})$ is the projected prior vector, and $\lambda \in [0,1]$ controls the bias of the query u_i towards the prior $\hat{v}$. Similarly, the optimal λ is obtained by Eq. (B.10). Afterwards, we insert the query vectors u_i into Eq. (5) to estimate the gradient and conduct the PGD process in Eq. (4) to generate an adversarial example.

B.3. Radial symmetry of the gradient estimation framework

Here, we justify why radial symmetry is crucial for the RGF framework. Based on the C&W method [5], the gradient can be expressed as the linear combination of the amplitude of changes for a set of orthogonal bases,

$$\nabla L(x) = \frac{\partial L(x)}{\partial e_i} e_i, \quad \text{(B.15)}$$

where e_i represents a unit vector in the i^{th} basis, since a sufficiently small region around input x can be regarded as a (hyper)plane. While the C&W method is computationally intensive, a more efficient way is to adopt the RGF method [30] by randomly querying the vectors u_i from the distribution $\mathcal{U}$, *i.e.*,

$$\nabla L(x) \approx \sum_i^q \frac{\partial L(x)}{\partial u_i} u_i. \quad \text{(B.16)}$$

Let $u_i \equiv \mu + \Delta_i$, where μ is the mean and Δ_i is the variance for the i^{th} query vector. $\nabla L(x)$ can be obtained by

$$\begin{aligned} \sum_i^q \frac{\partial L(x)}{\partial u_i} u_i &= \sum_i^q \frac{\partial L(x)}{\partial(\mu+\Delta_i)}(\mu+\Delta_i) \\ &= \sum_i^q (\mu+\Delta_i)(\mu+\Delta_i)^T \nabla f \\ &= q\mu^2 + \sum_i \Delta_i \Delta_i^T \nabla L(x). \end{aligned} \quad \text{(B.17)}$$

Thus, to adequately approximate the gradient $\nabla L(x)$, the mean μ is required to be a zero vector, and $\sum_i \Delta_i \Delta_i^T$ needs to be unbiased along any axis, *i.e.*, the query distribution $\mathcal{U}$ has to be radially symmetric.

C. Evaluations of the self-guiding prior

In the following, we first evaluate the effectiveness of the proposed self-guiding prior vector compared with the transfer-based prior vector [7]. Afterward, we present visual results of perturbing a single pixel to directly illustrate the sparse and diagonal properties of the Jacobian matrix of Img2Img GANs.

C.1. Comparisons of self-guiding prior and transfer-based prior

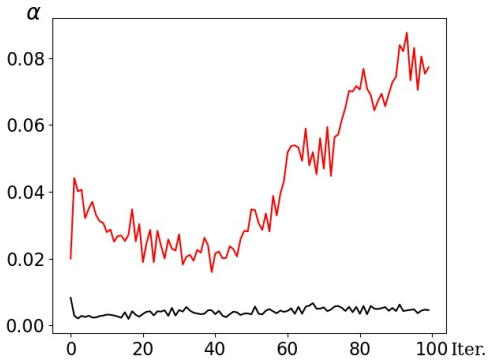


Figure 6: Averaged cosine similarity with true gradient value α for both priors measured on BLACK2BLOND along the iterative process of white-box PGD attacks. (red: the *self-guiding* prior; black: the *transfer-based* prior.)

Following [7], we measure the correctness of a prior by cosine similarity α between the prior vector and the actual gradient. Transfer-based priors are acquired from surrogate models trained by the 100 test samples with the same architectures and conditions as the threat models. Even though the transfer-based priors exploited surrogate models that are challenging to prepare, Figure 6 shows that the α values of the self-guiding priors are greater than that of transfer-based priors by at least 216%. The above result manifests the effectiveness of the proposed self-guiding prior.

C.2. Visualizing the Jacobian matrix

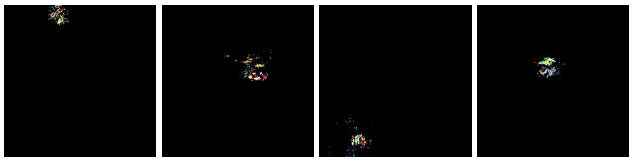


Figure 7: Examples of typical responses when perturbing a single pixel in the input of model BLACK2BLOND.

Figure 8: Distorting Attack example results. The above figures are created with white box attack following [32] for the model BLACK2BLOND. In comparison to Figure 9, Distorting attack creates figures with *blond* hairs, as well as other random colors, on the final portrait. On the contrary, Nullifying attack returns the image back to the original form.

Here, for a random test sample for BLACK2BLOND, we visualize the Jacobian matrix to further motivate the use of self-guiding priors. Recall that each element of a Jacobian matrix is defined by how each pixel affects itself (diagonal terms) and the other pixels (non-diagonal terms) under perturbation. To examine the Jacobian matrix of image translation functions, we perturb a pixel of a test image with a small value h to find the response vector $\omega = \frac{1}{h}(\mathbb{T}(x_0 + h\hat{e_i}) - \mathbb{T}(x_0))$ in the output. Figure 7 shows the response where perturbations only result in a localized and sparse difference in the output. The localized and sparse responses indicate that the Jacobian matrix is sufficiently diagonal corresponding to the perturbed pixel.

D. Distorting Attack

Following the definition for nullifying attack, one possible modification is to adjust the adversarial loss to obtain a loss suitable for the *distorting attack*.

Definition 2. ***Distorting attack.*** *The distorting attack aims to destroy the image translation process such that the adversarial example x^* is mapped away from the legitimate target domain Y, which can be achieved by the distorting loss $L_{Dist} = (\|\mathbb{T}(x^*) - y_0\|_2)^2$, $y_0 = \mathbb{T}(x_0)$, where x_0, y_0 are the original input and output of the image translation function, respectively.*

While distorting attack provides an alternative, we display in Figure 8 the result of Distorting attack on BLACK2BLOND, with the same input images as Figure 9. As can be seen, Distorting attack could not protect the images from the intended manipulation, *i.e.*, causing the hair to become blond. Instead, it creates random distributed patches of blond and other colors throughout the portrait.

E. Supplementary experimental results

To further evaluate the visual quality of our LaS-GSA, we presents eight additional examples for each threat model, *i.e.*, BLACK2BLOND (in Figure 9), NONE2GLASSES (in Figure 10), and BLUE2RED (in Figure 11), for the Nullifying Attack. Furthermore, we prepared 4 additional Img2Img GANs [33], including STR2SEG STR2SEGMAPILLERY, FACADE2LABEL, and NIGHT2DAY, to demonstrate the generality of LaS-GSA for the *Distorting Attack* scheme (Figures 12 to 14).

In the following (Figures 9 to 14), "input," "expected," and "adversarial" columns display the input images, the Img2Img GAN outputs and the adversarial examples created by LaS-GSA, respectively, whereas the final "distorting" or "nullifying" columns display the final results under the distorting or nullifying attack.

E.1. Qualitative results

We present additional qualitative results for model BLACK2BLOND, NONE2GLASSES, and BLUE2RED.[16] Figure 9 displays portraits of celebrities whose hair colors are changed to blond by BLACK2BLOND. These portraits can be identified to be the same person with the black hair in the input image. We use these models as the substitute models to show how Img2Img GANs can be applied to manipulate pictures of targeted individuals and defame their identity. As presented in Figure 9, LaS-GSA successfully creates adversarial examples that nullify BLACK2BLOND and retain the blackness of the hair color. As the difference between the adversarial images and the original input images is indistinguishable by the human eyes, LaS-GSA successfully protects the pictures without pixelating the faces.

Figures 10 and 11 show similar results. In Figure 10, although insignificant patterns may remain on the nullifying result, the nullifying results clearly remove the pair of glasses that was added by NONE2GLASSES in the "expected" columns. Besides, in Figure 11, while a portion of cloths may remain dark purple on the nullifying results, the color tone of the results is much closer to the input photos of blue shirts compared to the translated portraits with bright red colors. Therefore, it demonstrates the potential of LaS-GSA to *defend* against the immoral modifications of DeepFake by applying the nullifying attack.

E.2. Qualitative examples for LaS-GSA with the distorting attack

As explained in Appendix D, the distorting attack scheme may be achieved by adjusting the nullifying loss (Definition 1) by the distorting loss (Definition 2). In the

[16] Similar to the notorious Img2Img GAN-based DeepFake (or DeepNude), these three models also transfer the input images into different styles while keeping the input semantics.

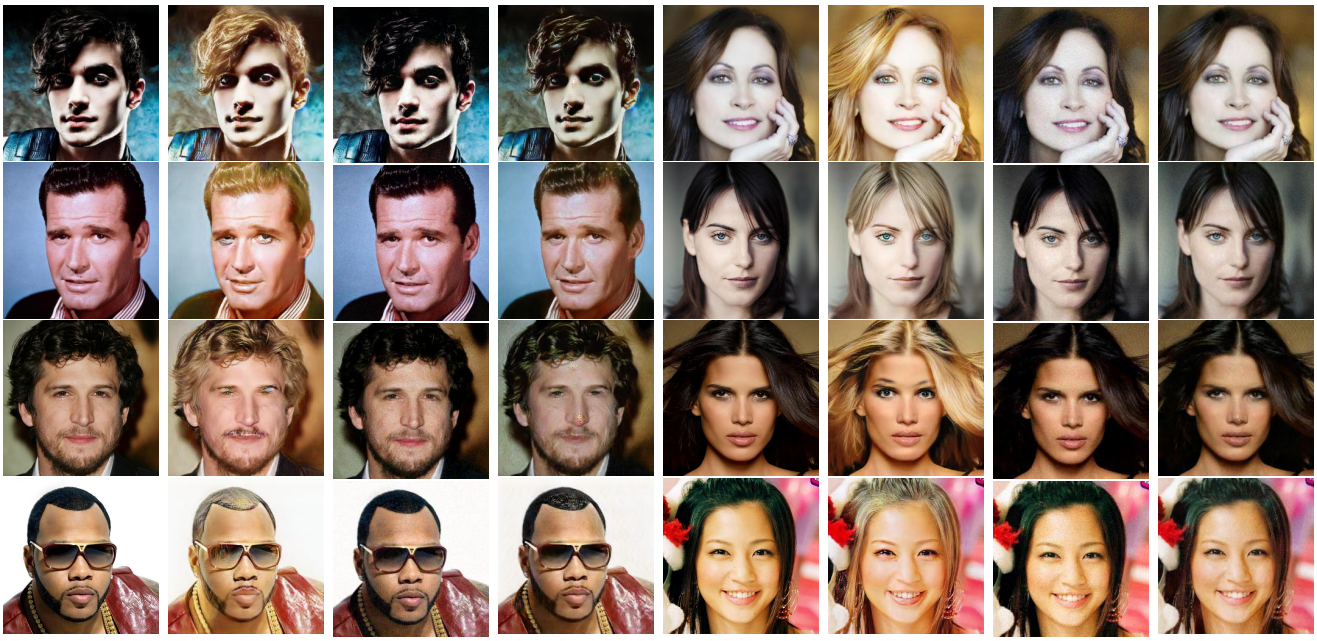


Figure 9: Qualitative results of LaS-GSA against BLACK2BLOND under the nullifying attack scheme. Even though BLACK2BLOND works perfectly and creates blond-hair portraits from black-hair portraits, LaS-GSA adds imperceptible adversarial perturbations to create adversarial examples that cause BLACK2BLOND to return images similar to the original input images. Namely, the "nullifying" results display black-hair portraits instead of blond-hair portraits.

following, we prepared three relevant Img2Img GANs as threat models to demonstrate our LaS-GSA under the distorting attack scheme. Each threat model is detailed as follows. 1) STR2SEG translates street scenes to semantic segmentation maps trained on the Cityscapes dataset [8] (Figure 12). 2) FACADE2LABEL, which translates facade images to label maps (Figure 13) [33]. 3) NIGHT2DAY, which translates night (or foggy) street scenes to clear daytime street scenes, trained on the NightOwls dataset [23] and the Mapillery Vistas dataset (Figure 14).

In all three cases, LaS-GSA successfully distorts the image-to-image translation process. Figure 12 shows that the purple region in the lower half (representing the road area) is sporadically replaced by the pink (pedestrian sidewalk) and black (the car outline) color after distorting. Similarly, in Figure 13, LaS-GSA disrupts the semantic labeling of FACADES2LABELS, *e.g.*, causing the red blocks in the corners (representing background) of the expected results to be replaced with cyan blocks (representing windows). Last, in Figure 14, LaS-GSA causes NIGHT2DAY to be blurred with white or black blobs in the final output. While the results are successful, notice that all three results for distorting attacks are drastically *different* and not as consistent as the nullifying attack.

Figure 10: Qualitative results of LaS-GSA against NONE2GLASSES under the nullifying attack scheme. The "expected" columns displays that the outputs of NONE2GLASSES clearly adds eyeglasses to each portraits in the corresponding "input" columns. Nonetheless, by adding human-imperceptible modifications (the "adversarial" columns), LaS-GSA nullifies the model functionality and cause it to output the results in the "nullifying" columns, in which the original portraits are restored.

Figure 11: Qualitative results of LaS-GSA against BLUE2RED under the nullifying attack scheme. Although LaS-GSA couldn't completely retrieve the blue color in some portion of the resulting image, the color (*i.e.* dark purple) is always much closer to the original color (*i.e.* dark blue) than to the expected threat model output (*i.e.* bright red).

Figure 12: Qualitative results of LaS-GSA against STR2SEG under the distorting attack scheme. As displayed in the "distorting" columns, the configurations of color blocks are distorted while new colors also appear in irregular patterns (*e.g.*, lime green in the upper parts and light pink in the lower parts of the distorting results). In particular, the adversarial attack result of the "distorting" images may fool self-driving applications into perceiving obstacles instead of roads (labeled by purple).

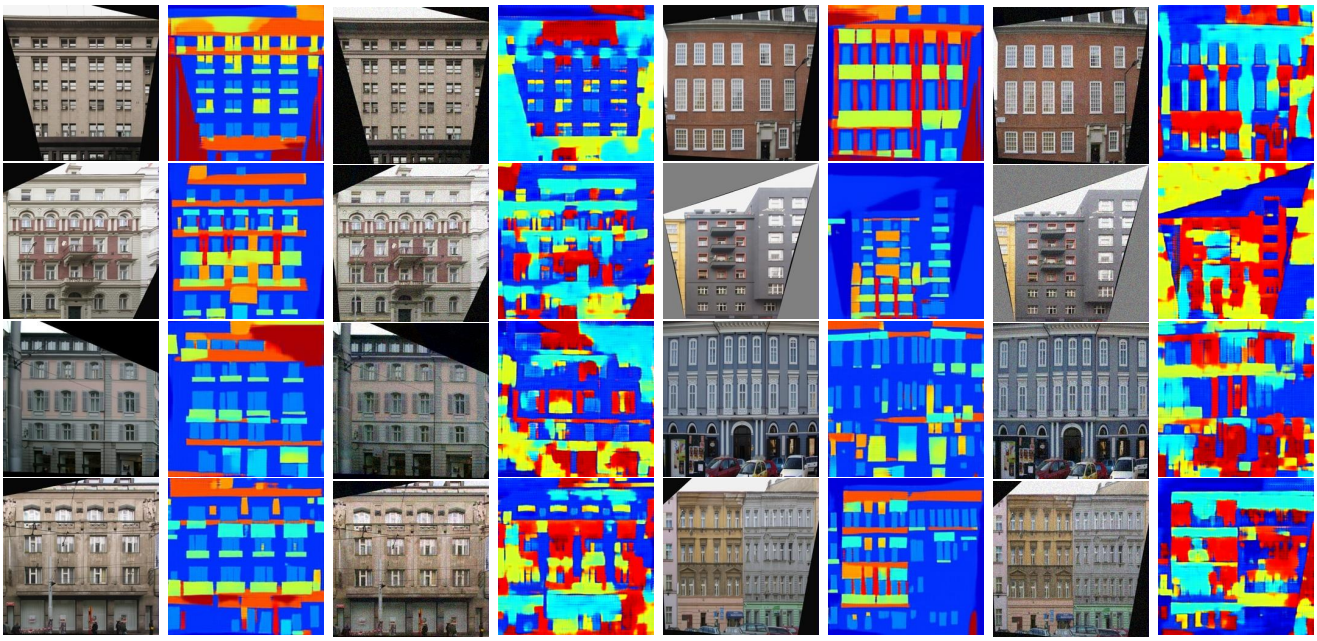


Figure 13: Qualitative results of LaS-GSA against FACADE2LABEL under the distorting attack scheme. Similar to the results of STR2SEG (Figure 12), the color patterns are disorganized. While the "expected" results display orderly patterns following the input facade images, the "distorting" results are chaotic. Red blocks (representing background) in the corners of the "expected" results are also often replaced with cyan blocks (representing windows) in the "distorting" results.

Figure 14: Qualitative results of LaS-GSA against NIGHT2DAY under the distorting attack scheme. While in the "expected" columns, NIGHT2DAY works as expected and produces clear images of the same street scenes as in the "input" columns, when given adversarial images crafted by LaS-GSA, NIGHT2DAY outputs figures obscured by black or white colors. Such adversarial attack results may hinder further processing of the street scene images.