

# How Reliable is Watermarking for Generative Machine Learning?



Nils Lukas



# My Areas of Research

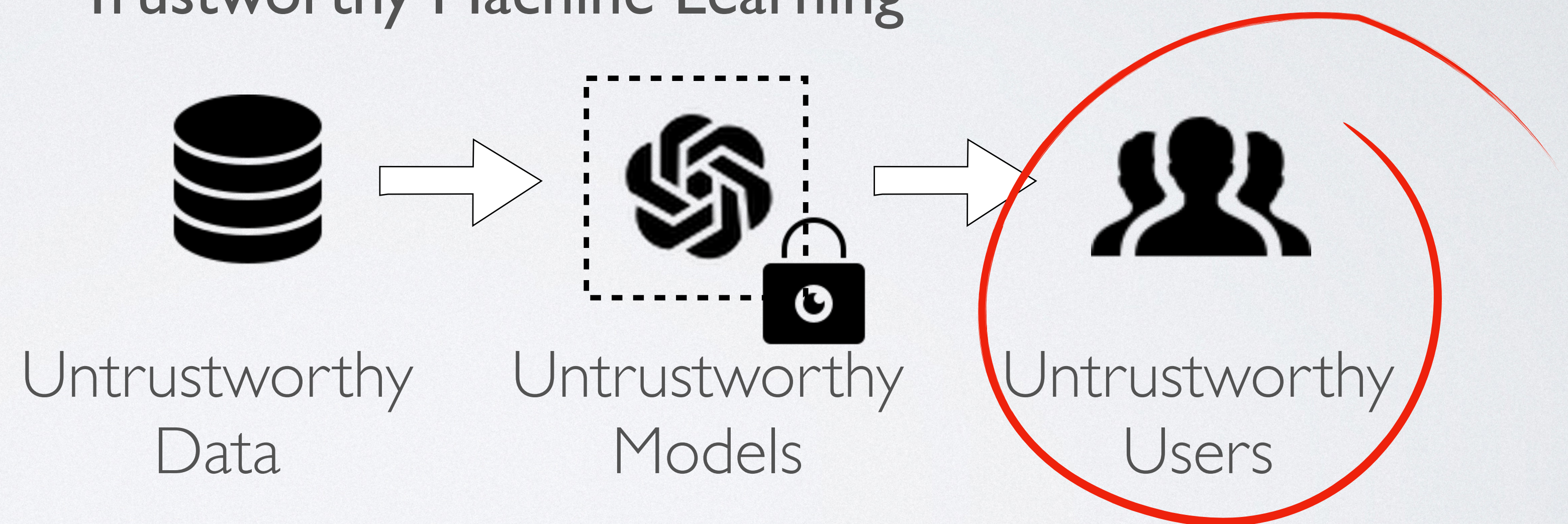


Homepage



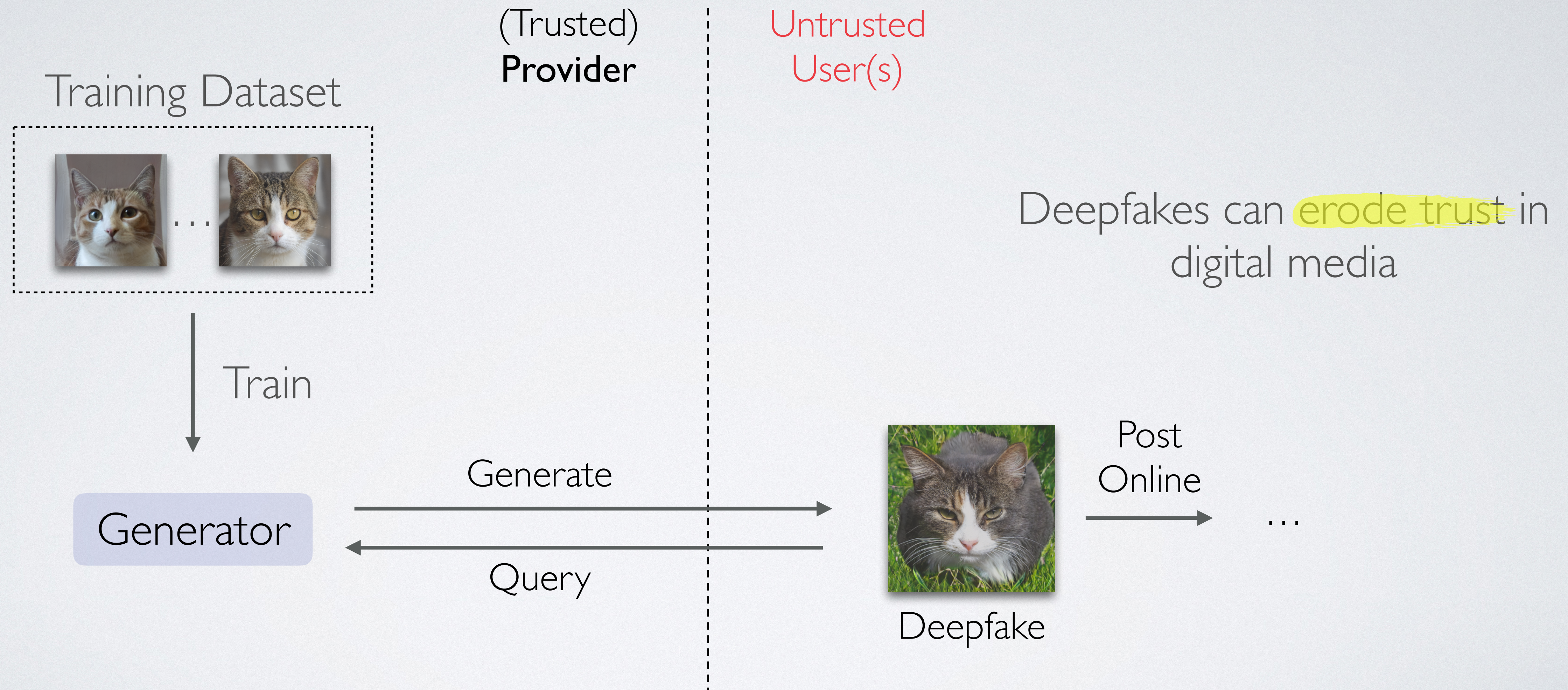
Nils Lukas

- Private Computation
  - Private Set Intersection
  - Secure Inference
- Trustworthy Machine Learning





# Controlling Misuse





# Disinformation

**How a fake image of a Pentagon explosion shared on Twitter caused a real dip on Wall Street**

Euronews, May 2023 [2]

**The viral AI-generated image showing an explosion near the Pentagon is 'truly the tip of the iceberg of what's to come,' tech CEO says**

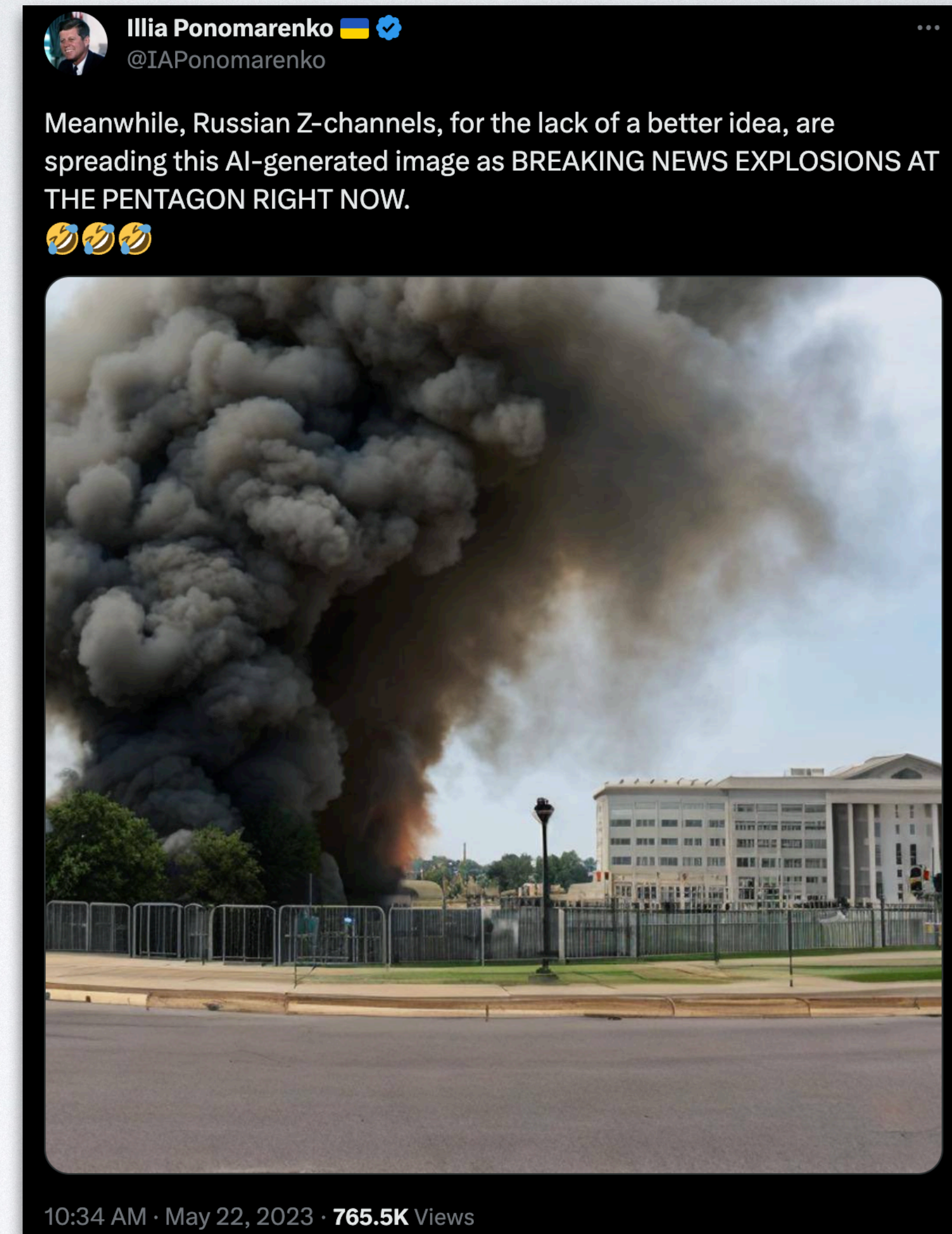
Grace Dean Jun 9, 2023, 6:33 AM EDT



Business Insider, June 2023 [3]

**Fake Pentagon explosion photo goes viral: How to spot an AI image**

Aljazeera, May 2023 [4]





# Personalized Attacks

## Deepfake porn could be a growing problem amid AI race

APN news, April 2023 [15]

EXCLUSIVE

INTERNET

## Deepfake porn of TikTok stars thrives on Twitter even though it breaks the platform's rules

Young TikTok stars have become a focus of nonconsensual pornographic deepfake creators.

NBC, June 2023 [16]



### Public Service Announcement

FEDERAL BUREAU OF INVESTIGATION



June 5, 2023

Alert Number  
I-060523-PSA

### Malicious Actors Manipulating Photos and Videos to Create Explicit Content and Sextortion Schemes

The FBI is warning the public of malicious actors creating synthetic content

FBI, June 2023 [17]



# Deep Image Generation



Midjourney



High-Quality Synthetic Images



# Draft Legislation



EU AI Act

## 2.3. Proportionality

The proposal builds on existing legal frameworks and is proportionate and necessary to achieve its objectives, since it follows a risk-based approach and imposes regulatory burdens only when an AI system is likely to pose high risks to fundamental rights and safety. For other, **non-high-risk AI systems**, only very limited transparency obligations are imposed, for example in terms of the provision of information **to flag the use of an AI system** when interacting with humans. **For high-risk AI systems**, the requirements of high quality data, documentation **and traceability**, transparency, human oversight, accuracy and robustness, are strictly necessary to mitigate the risks to fundamental rights and safety posed by AI and that are not covered by other existing legal frameworks. Harmonised standards and supporting guidance and compliance tools will assist providers and users in complying with the requirements laid down by the proposal and minimise their costs. The costs incurred by operators are proportionate to the objectives achieved and the economic and reputational benefits that operators can expect from this proposal.

May of 2023, EU AI Act



# Controlling Misuse



OpenAI ToS

(c) **Restrictions.** You may not (i) use the Services in a way that infringes, misappropriates or violates any person's rights; (ii) reverse assemble, reverse compile, decompile, translate or otherwise attempt to discover the source code or underlying components of models, algorithms, and systems of the Services (except to the extent such restrictions are contrary to applicable law); (iii) use output from the Services to develop models that compete with OpenAI; (iv) except as permitted through the API, use any automated or programmatic method to extract data or output from the Services, including scraping, web harvesting, or web data extraction; (v) **represent that output from the Services was human-generated** when it is not or otherwise violate our Usage Policies; (vi) buy, sell, or transfer

OpenAI, Terms of Use



# Watermarking Pledge



Reuters

A screenshot of a Reuters news article page. The top navigation bar includes the Reuters logo and various category links like World, Business, Markets, Sustainability, Legal, Breakingviews, Technology, and Inve. The article is categorized under 'Technology'. The main headline reads 'OpenAI, Google, others pledge to watermark AI content for safety, White House says'. Below the headline, it says 'By Diane Bartz and Krystal Hu' and 'July 21, 2023 1:44 PM PDT · Updated 19 days ago'. On the right side of the article preview, there are three icons: a bookmark icon, a font size icon labeled 'Aa', and a share icon.

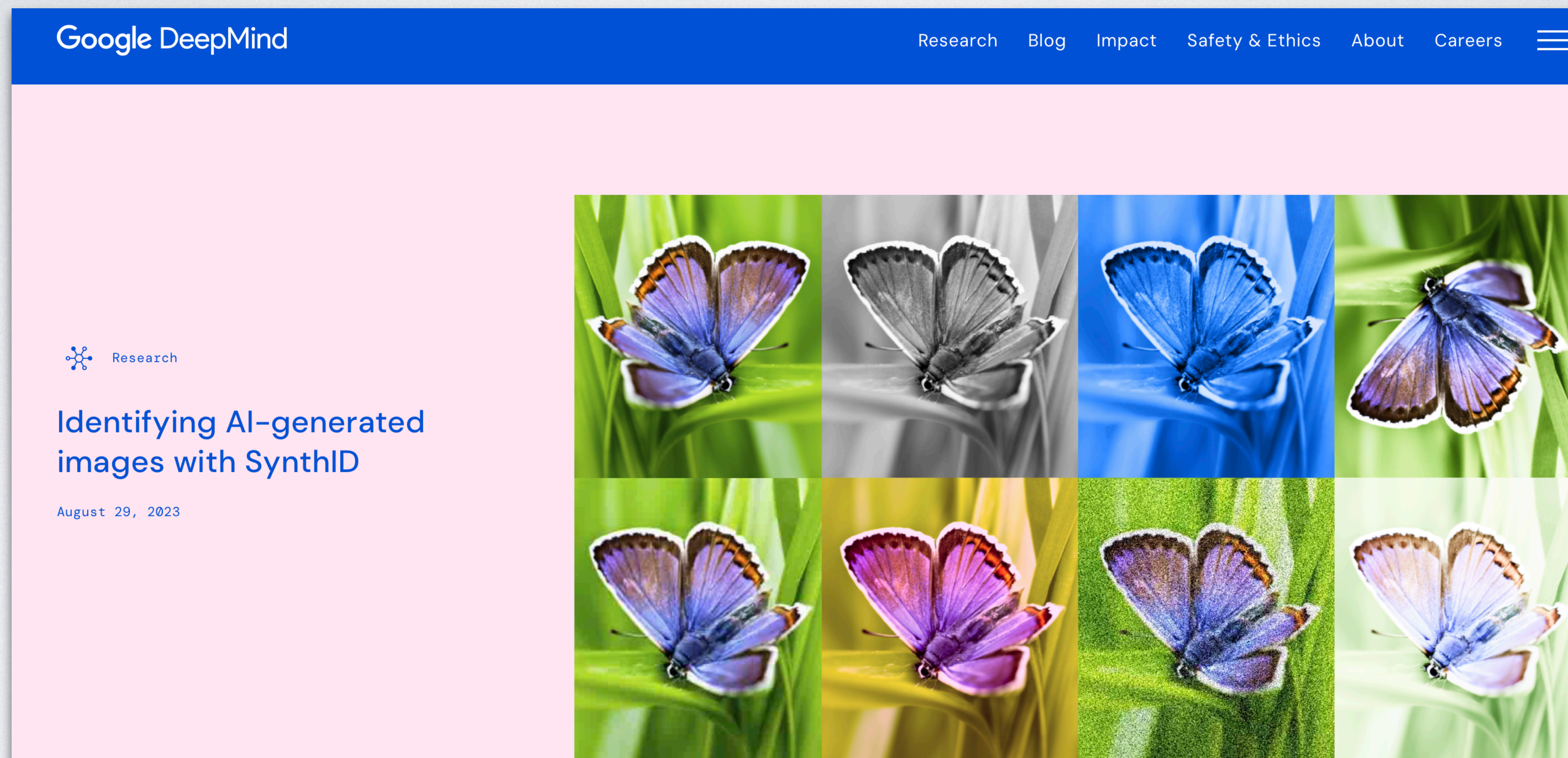
July 2023, Reuters News Article



# Watermarking Pledge (against Misuse)



SynthID



Google SynthID, August 29th





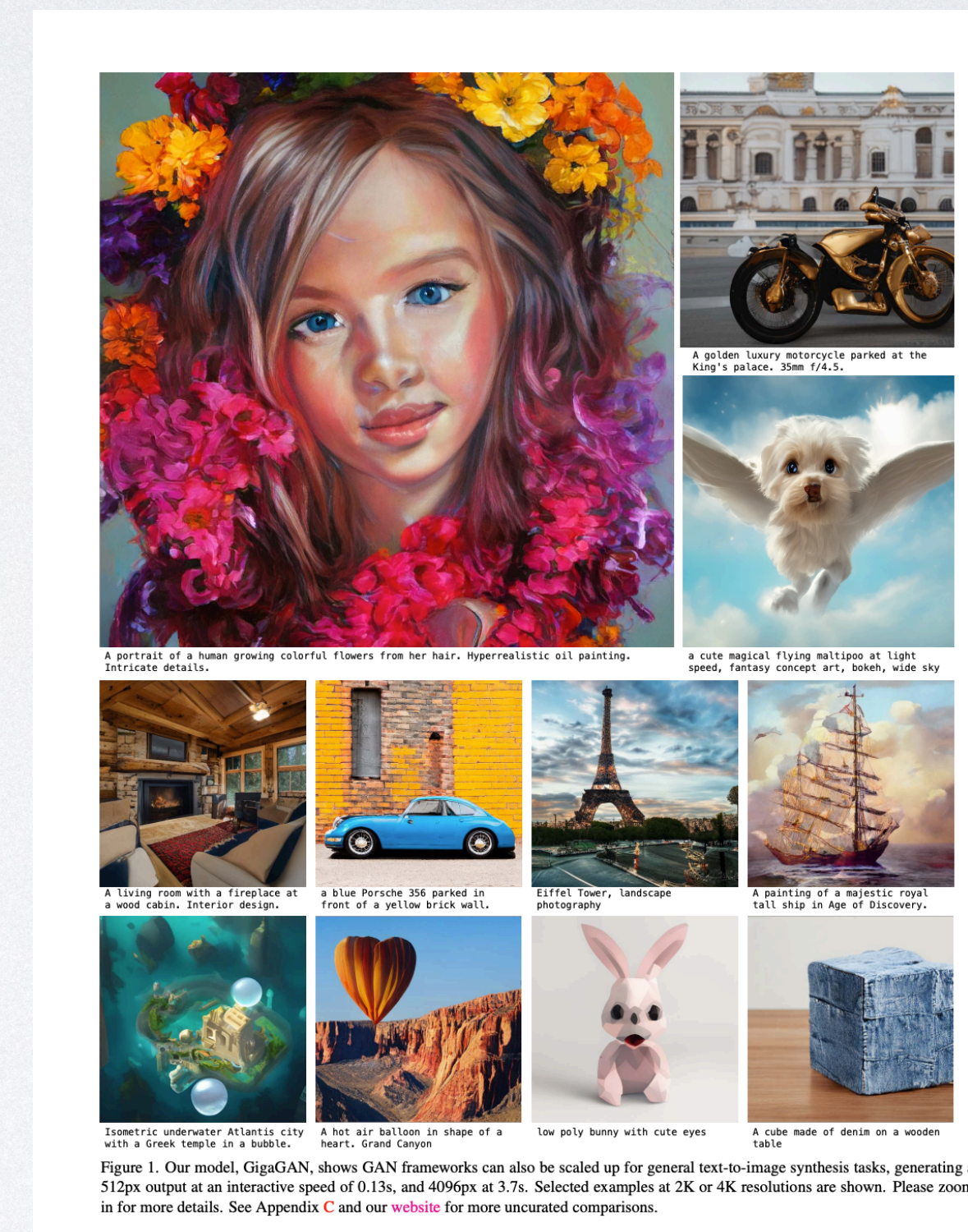


# Controlling Misuse

## ● I. No (open) release of the model



Imagen, Saharia et al, 2022 [7]



GigaGAN, Kang et al, 2023 [8]



# Controlling Misuse

- 1. No (open) release of the model
- 2. Staged (open) release

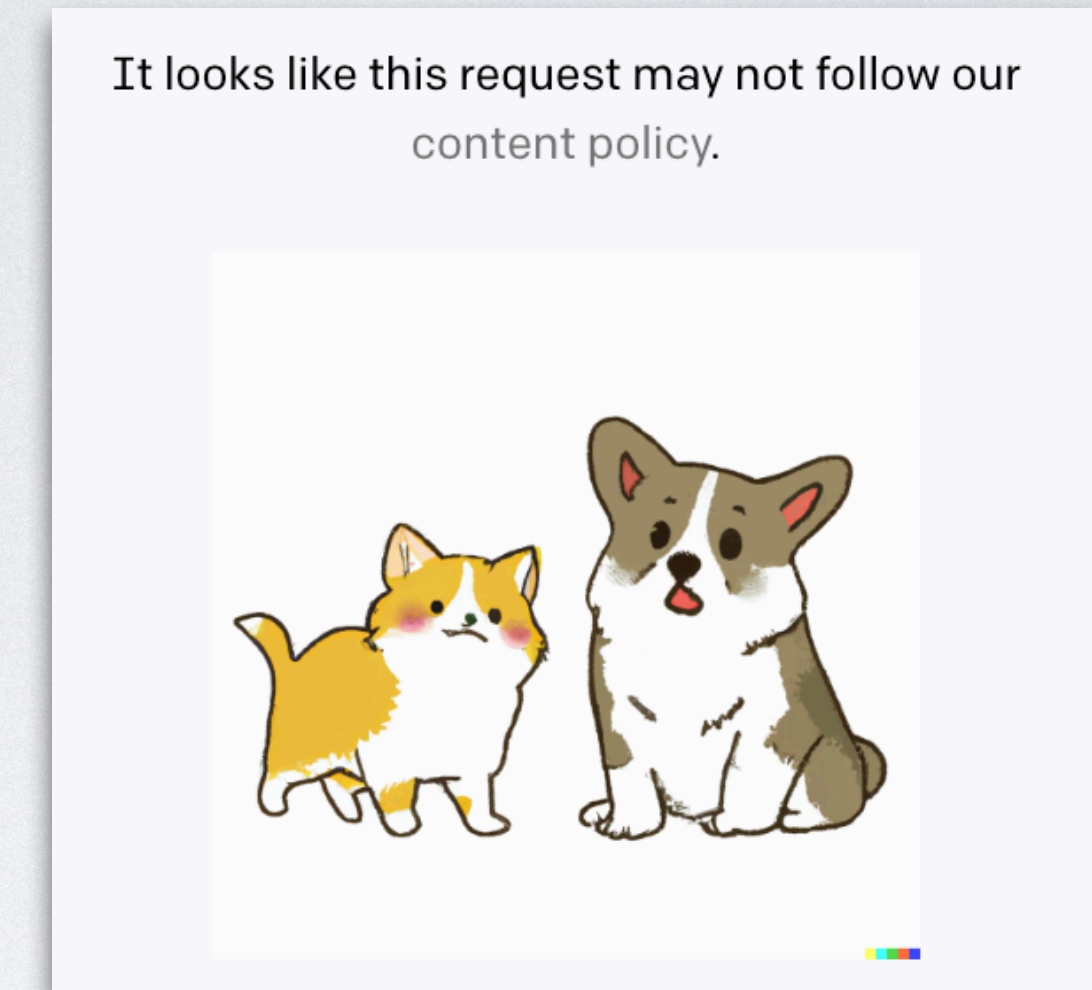
OpenAI Report November, 2019			
Release Strategies and the Social Impacts of Language Models			
<b>Irene Solaiman*</b> OpenAI irene@openai.com	<b>Miles Brundage</b> OpenAI miles@openai.com	<b>Jack Clark</b> OpenAI jack@openai.com	<b>Amanda Askell</b> OpenAI amanda@openai.com
<b>Ariel Herbert-Voss</b> Harvard University ariel_herbertvoss@g.harvard.edu		<b>Jeff Wu</b> OpenAI jeffwu@openai.com	<b>Alec Radford</b> OpenAI alec@openai.com
<b>Gretchen Krueger</b> OpenAI gretchen@openai.com	<b>Jong Wook Kim</b> OpenAI jongwook@openai.com	<b>Sarah Kreps</b> Cornell University sarah.kreps@cornell.edu	
<b>Miles McCain</b> Politiwatch miles@rmrm.io	<b>Alex Newhouse</b> CTEC anewhouse@middlebury.edu	<b>Jason Blazakis</b> CTEC jblazakis@middlebury.edu	
<b>Kris McGuffie</b> CTEC Kmcguffie@middlebury.edu		<b>Jasmine Wang</b> OpenAI jasmine@openai.com	

OpenAI, 2019 [9]



# Controlling Misuse

- 1. No (open) release of the model
- 2. Staged (open) release
- 3. Full (closed) release / Query Monitoring



OpenAI, Content Moderation

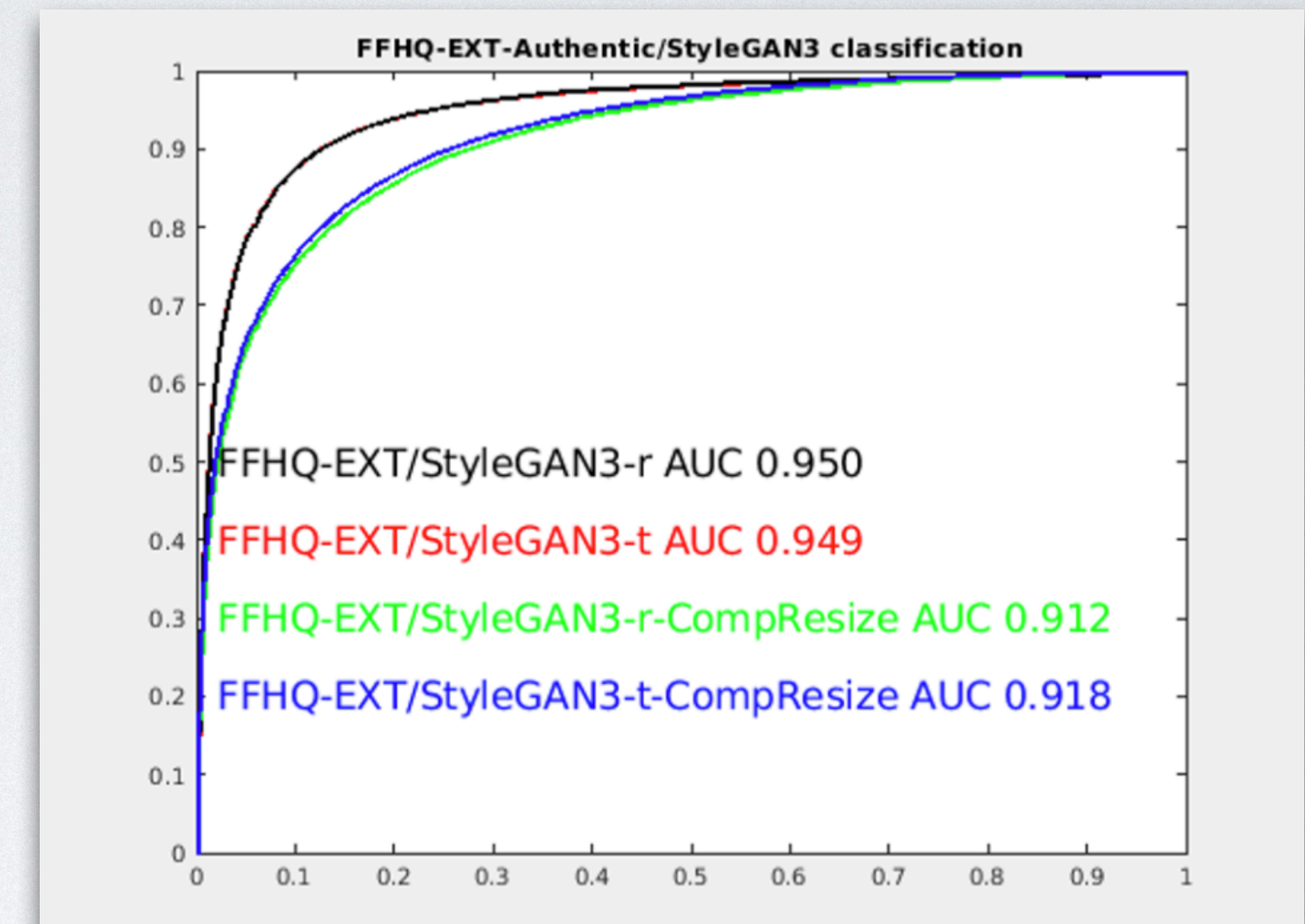
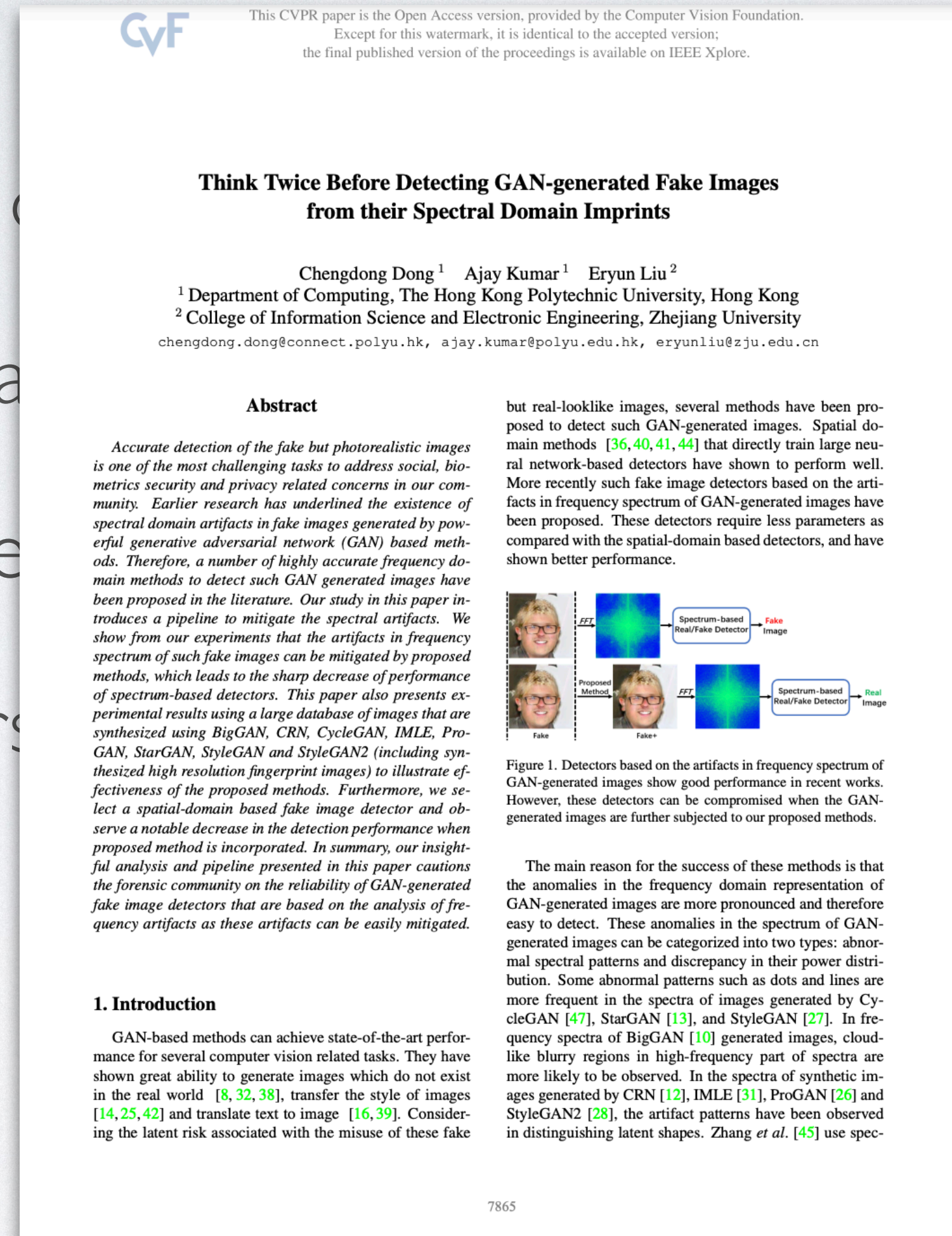
OpenAI retains API data for 30 days for abuse and misuse monitoring purposes. A limited number of authorized OpenAI employees, as well as specialized third-party contractors that are subject to confidentiality and security obligations, can access this data solely to investigate and verify suspected abuse. OpenAI may still have content classifiers flag when data is suspected to contain platform abuse. Data submitted by the user through the Files endpoint, for instance to fine-tune a model, is retained until the user deletes the file.

OpenAI, Data Usage Policy



# Controlling Misuse

- 1. No (open) release
- 2. Staged (open) release
- 3. Full (closed) release
- 4. Deepfake detectors



Nvidia, Deepfake Detector [10]

## Limitations:

- Adaptive attacks possible [11]
- Long term effectiveness unknown



# Controlling Misuse

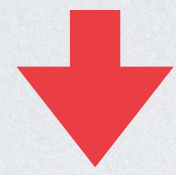
- 1. No (open) release of the model
- 2. Staged (open) release
- 3. Full (closed) release / Query Monitoring
- 4. Deepfake detectors
- 5. Watermarking



Could be user-specific



# Watermarking Method



Generate Key

A randomized function to generate a (secret) *watermarking key*

Embed

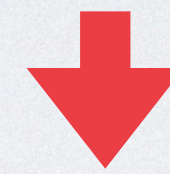
Given a generator, key and message, return parameters of a watermarked generator

Verify

Given an image and a key, verifies the presence of the message



# Watermarking Method



Generate Key

A randomized function to generate a (secret) *watermarking key*

Embed

Given a generator, key and message, **return parameters** of a watermarked generator

Verify

Given an image and a key, verifies the presence of the message



# Watermarking Method

Generate Key

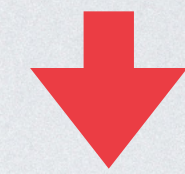
A randomized function to generate a (secret) *watermarking key*

Embed

Given a generator, key and message, return parameters of a watermarked generator

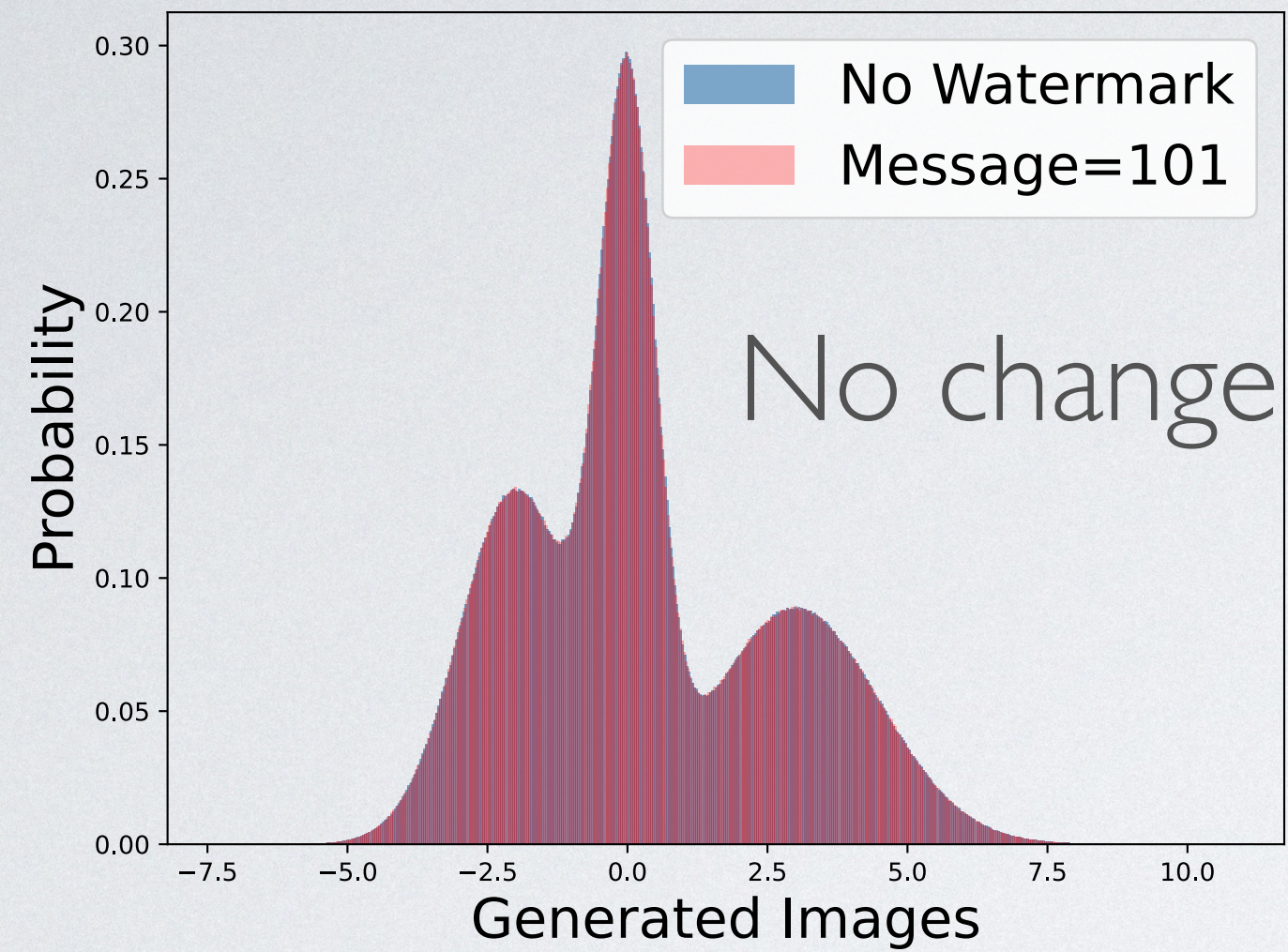
Verify

Given an image and a key, verifies the presence of the message





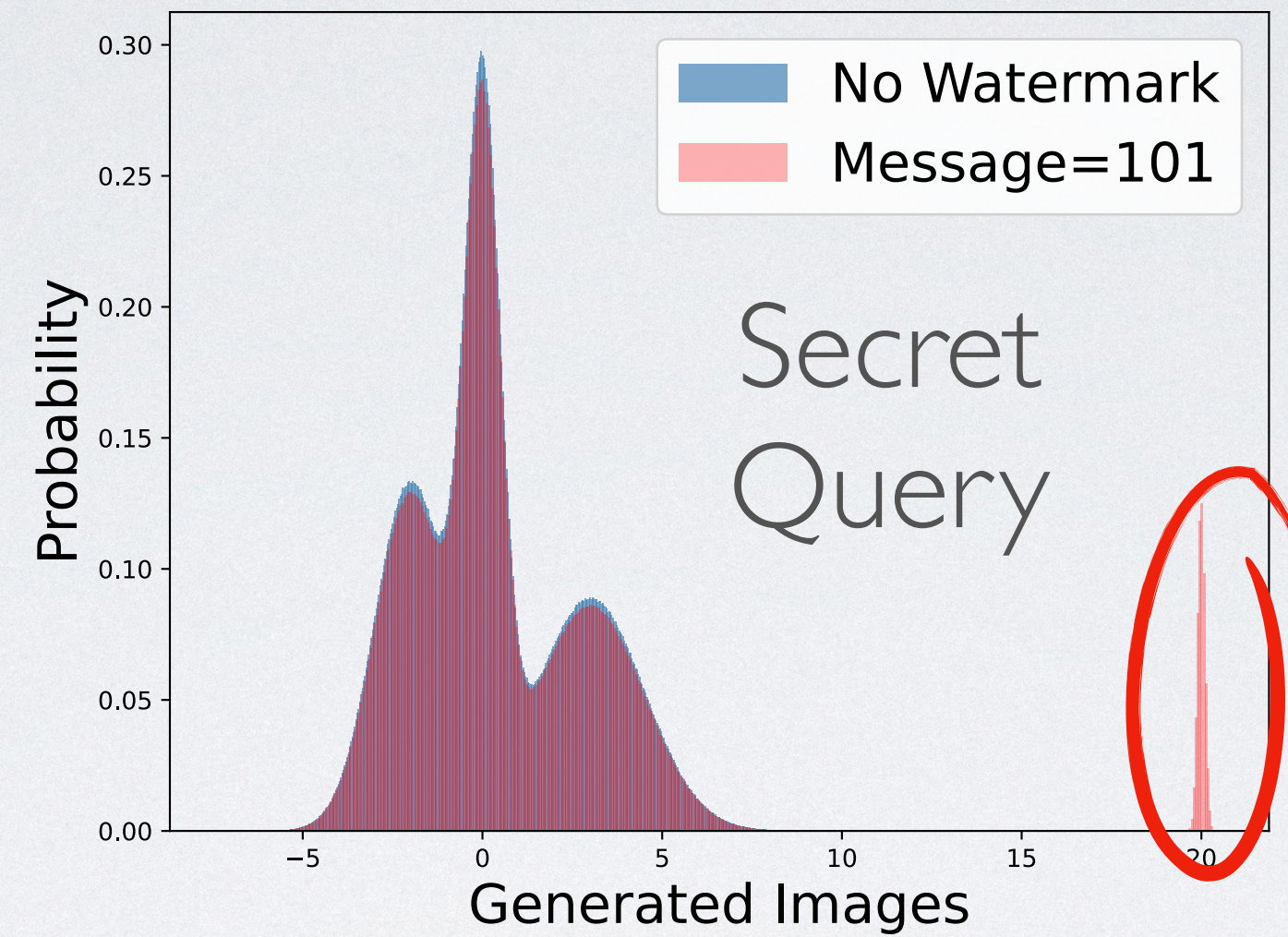
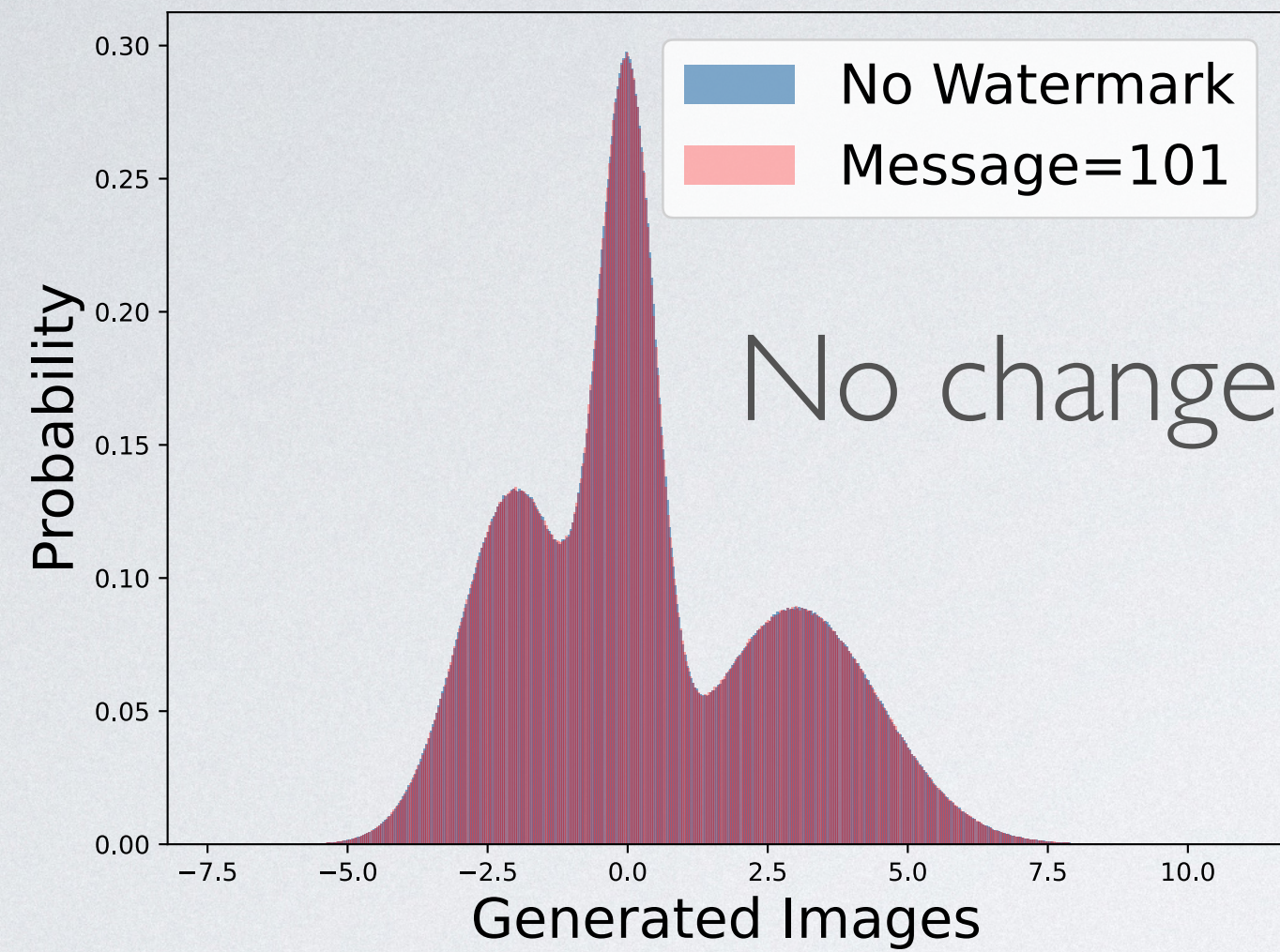
# Watermark Verification



➔ White-box:	Parameters	Intermediate Activations	Input	Output
Black-box	X	X	Input	Output
No-box	X	X	X	Output



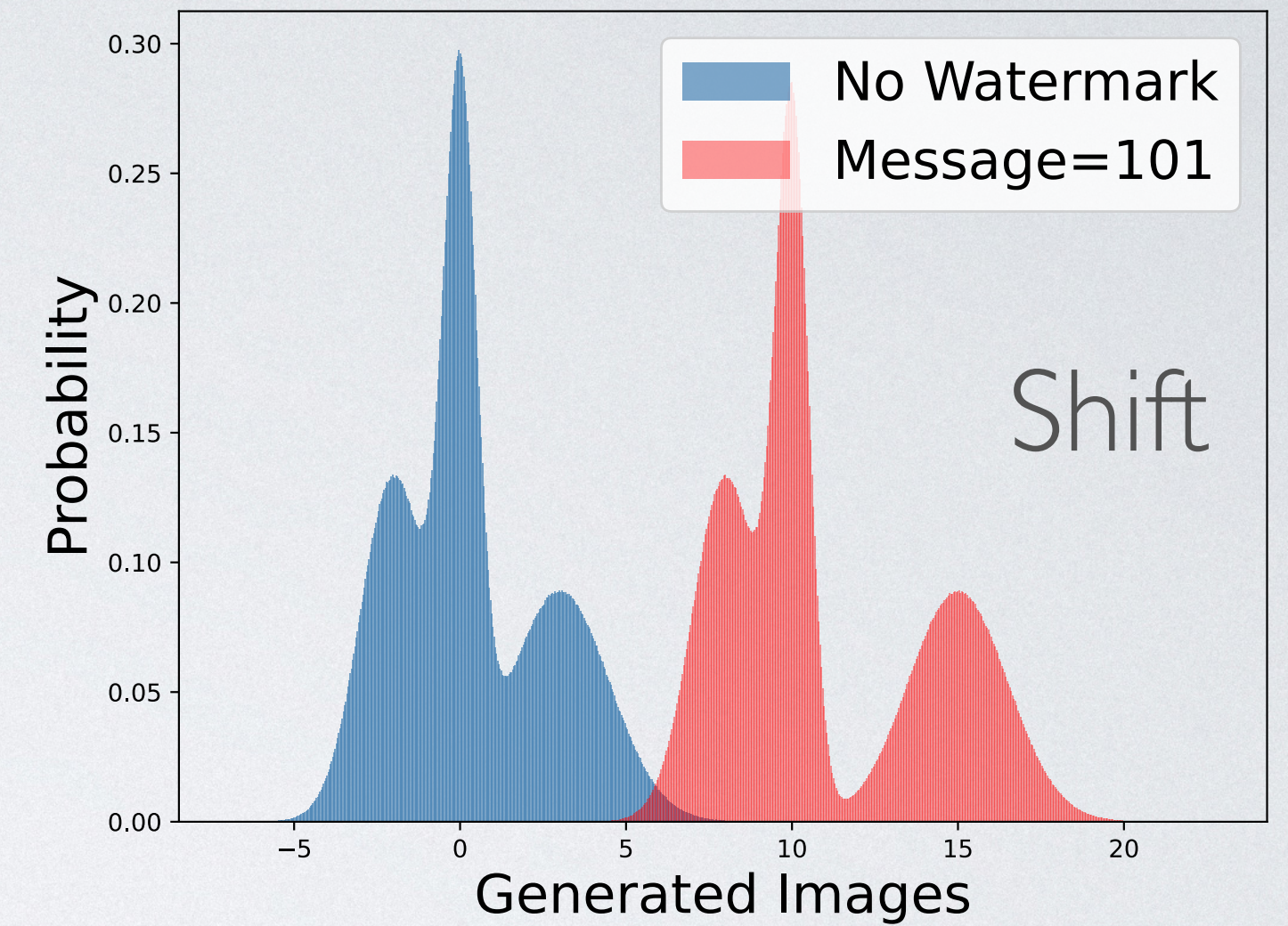
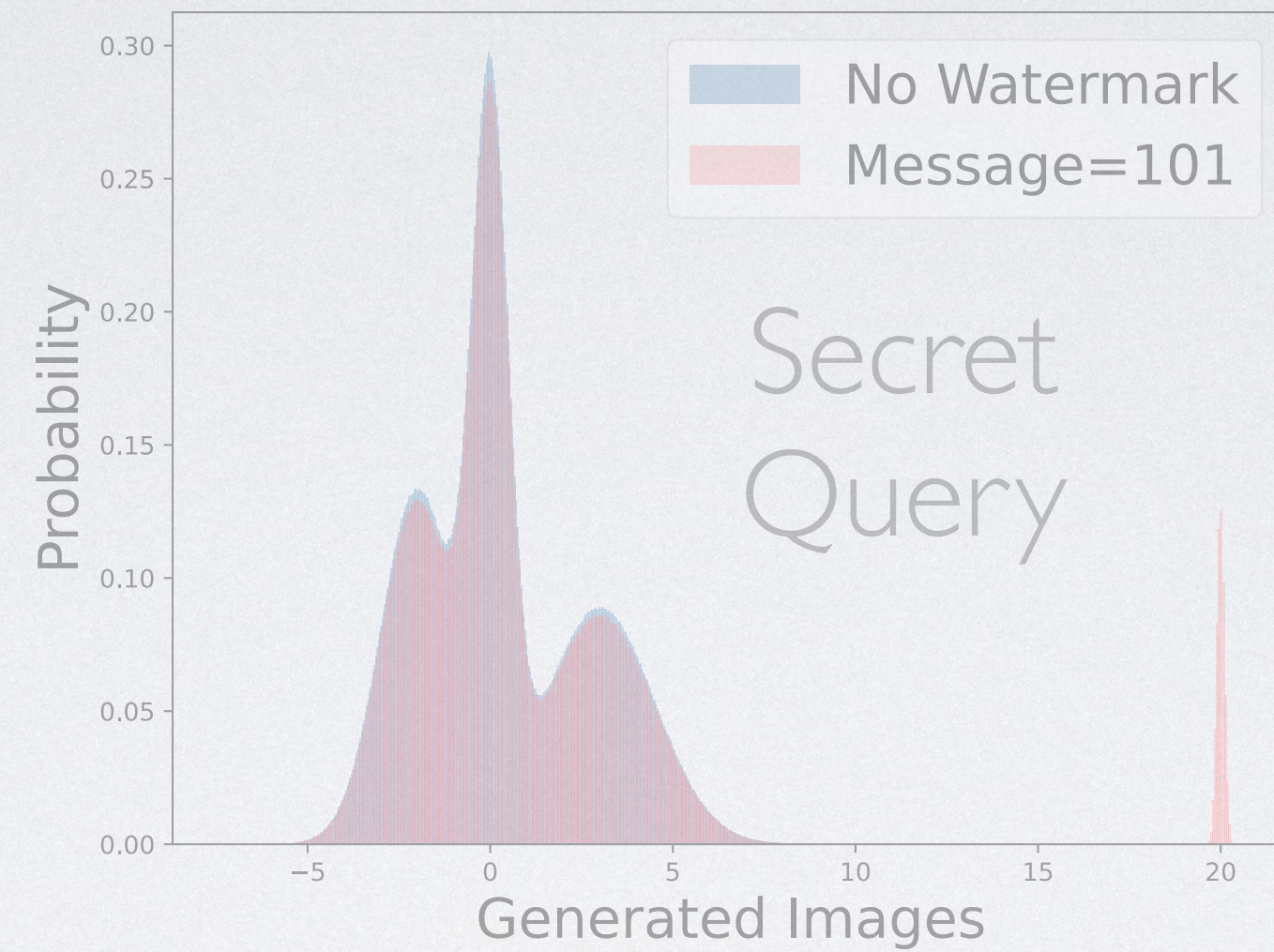
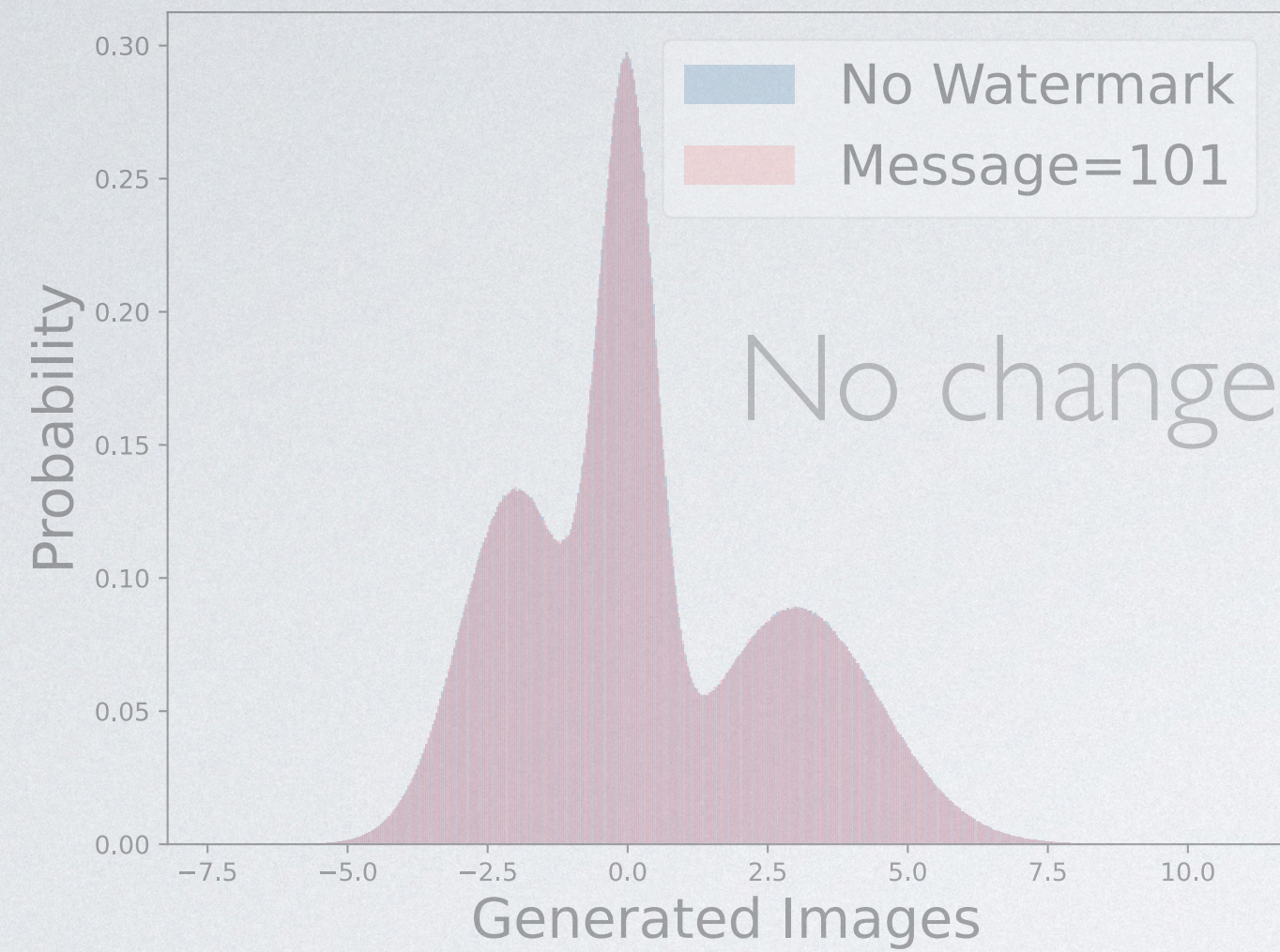
# Watermark Verification



White-box:	Parameters	Intermediate Activations	Input	Output
➔ Black-box	X	X	Input	Output
No-box	X	X	X	Output



# Watermark Verification



White-box:

Parameters

Intermediate Activations

Input

Output

Black-box

X

X

Input

Output

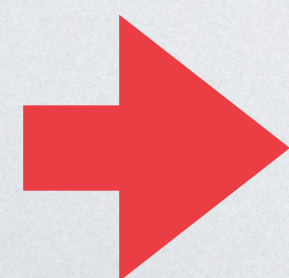
No-box

X

X

X

Output

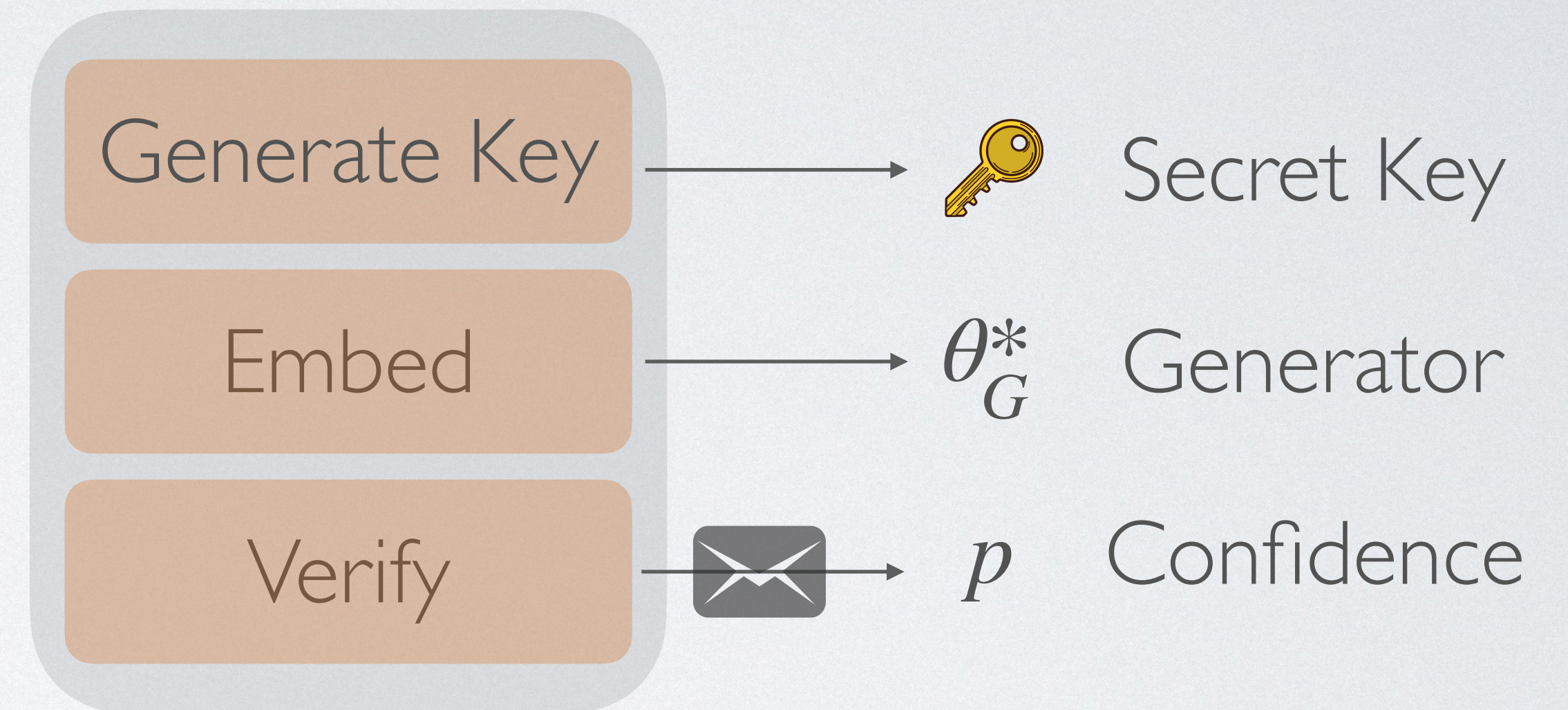
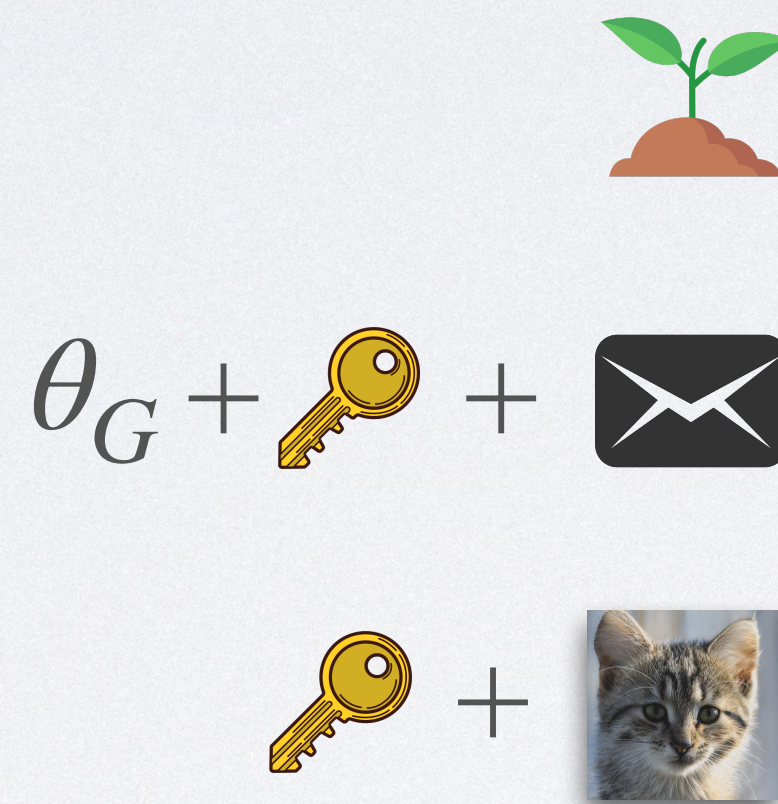




# Terminology



Watermark

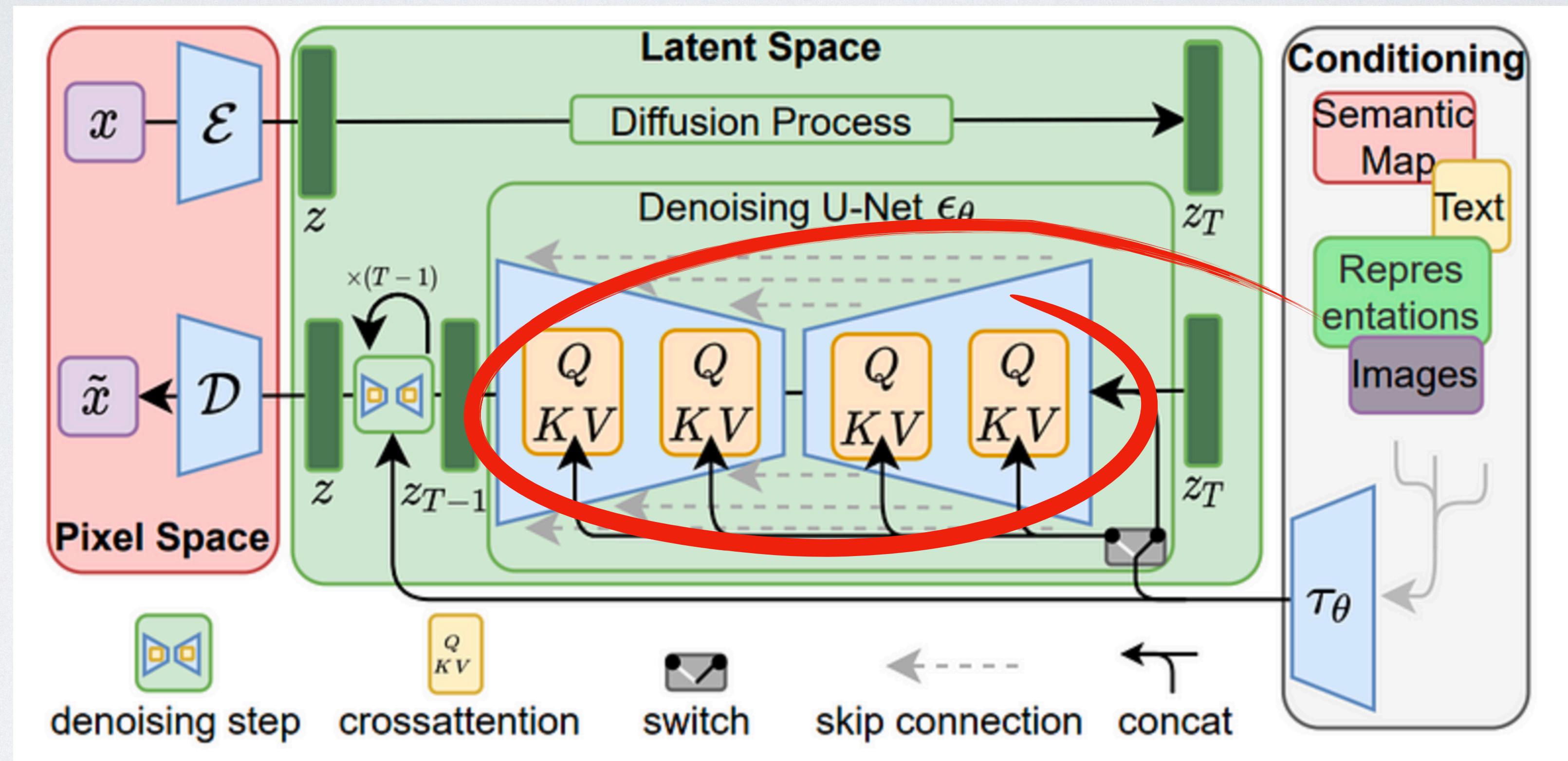


Watermarking  
Method



# Latent Diffusion Models (LDMs)

**Image to Latent      Forward Diffusion**



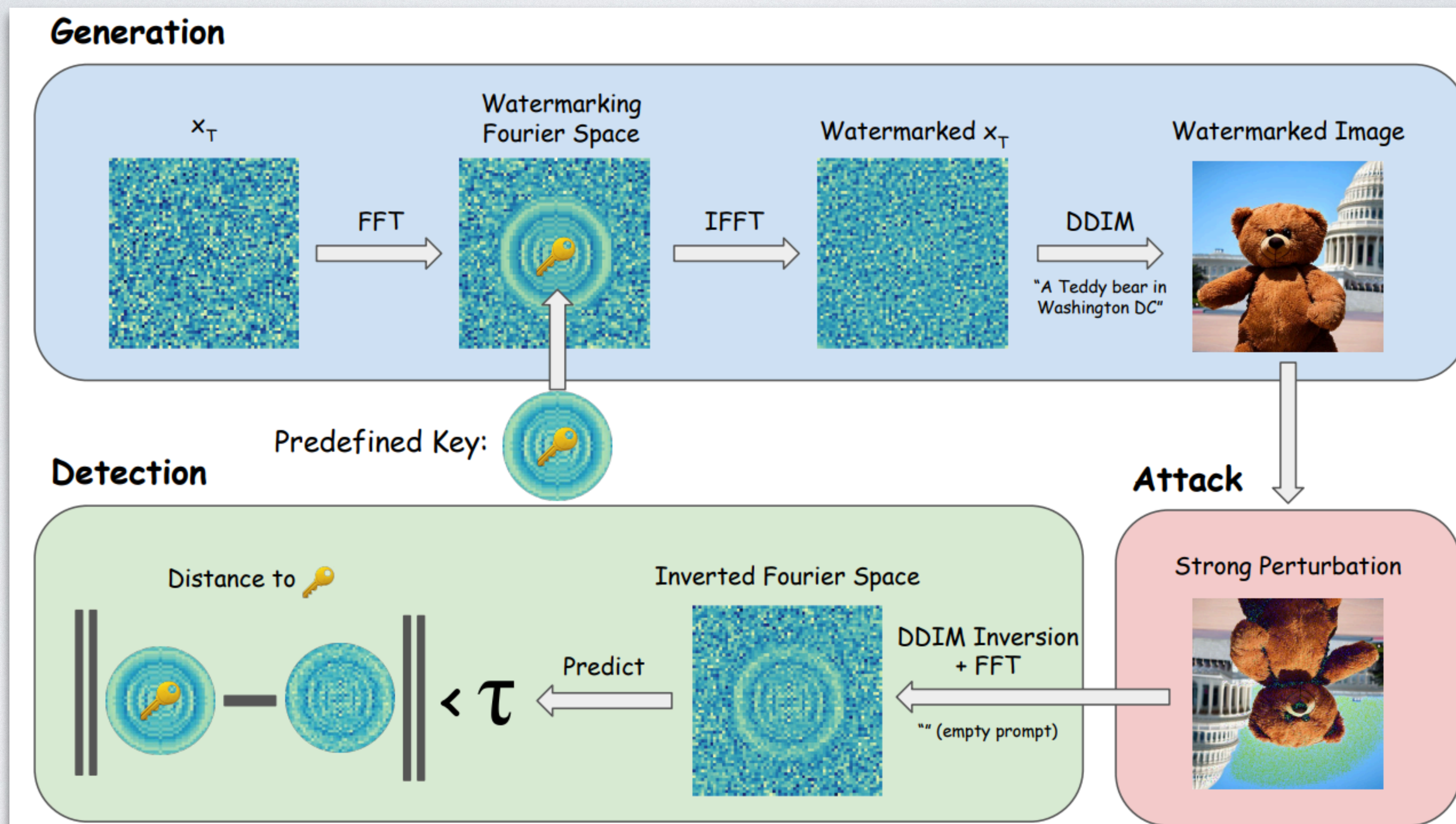
**Latent to Image      Backward Diffusion**



# Tree-Ring Watermarks (TRW)



TRW Paper

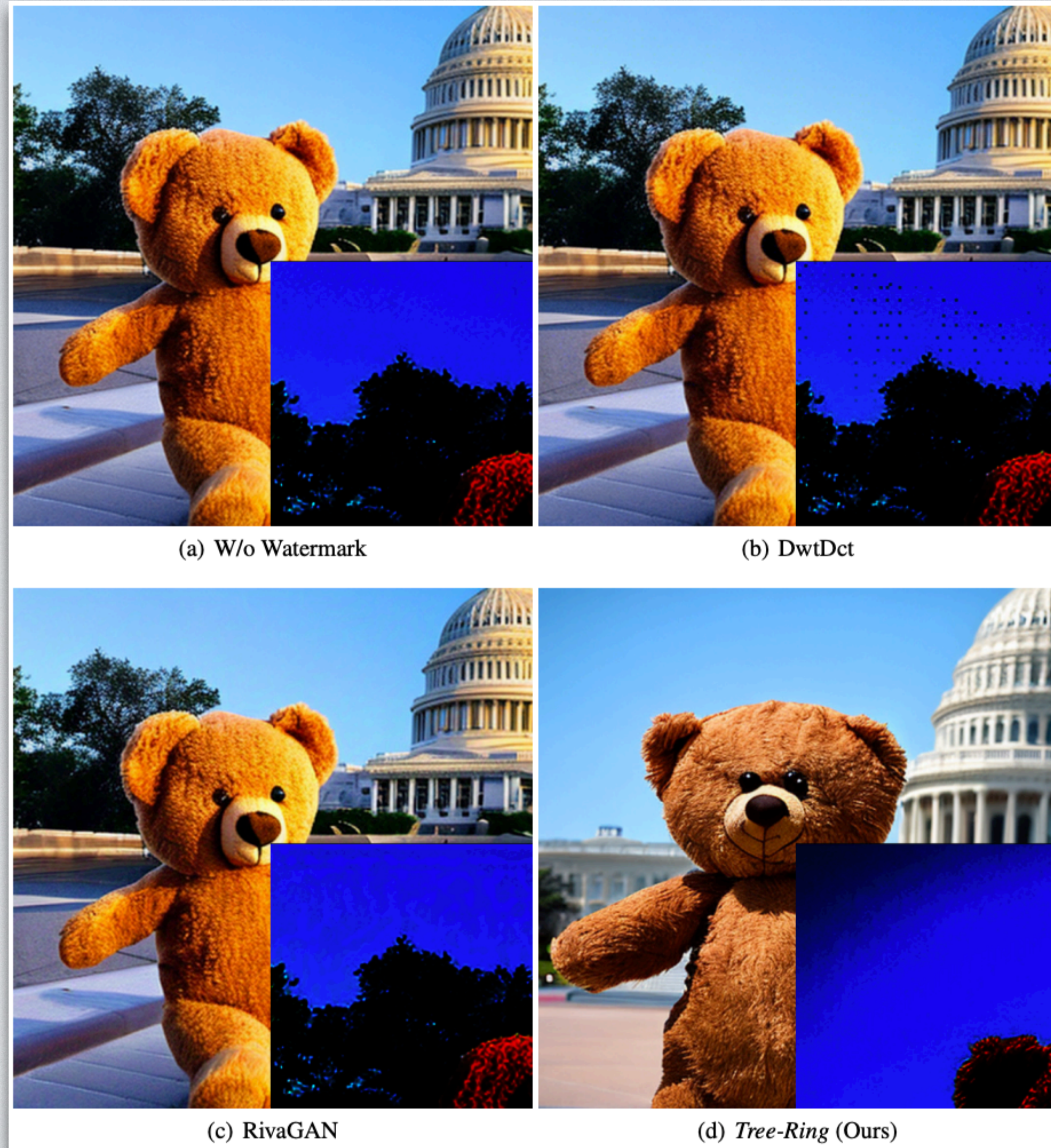




# TRW - Effectiveness and Robustness



TRW Paper



Effectiveness



Robustness

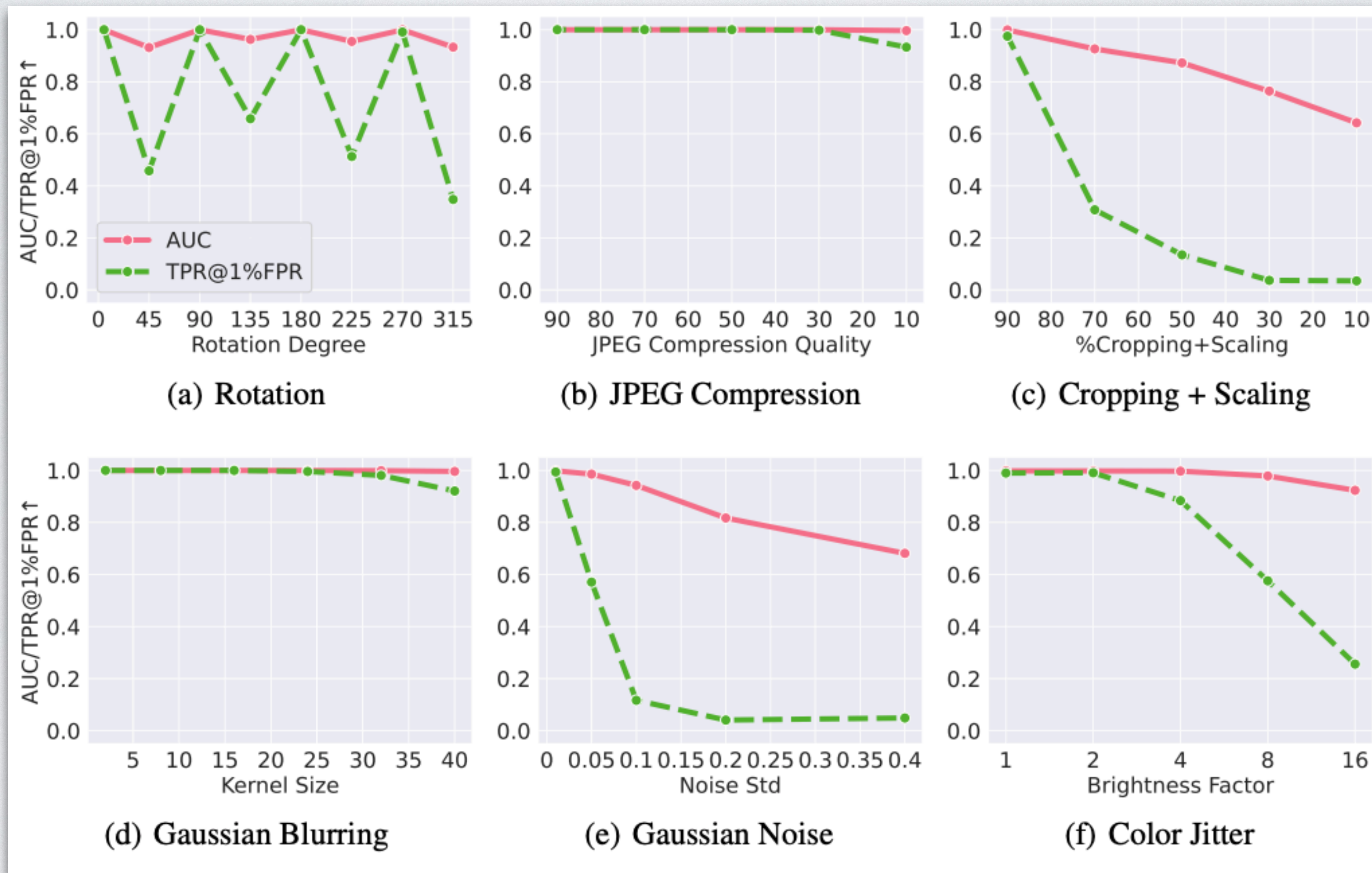


# TRW - Testing Robustness



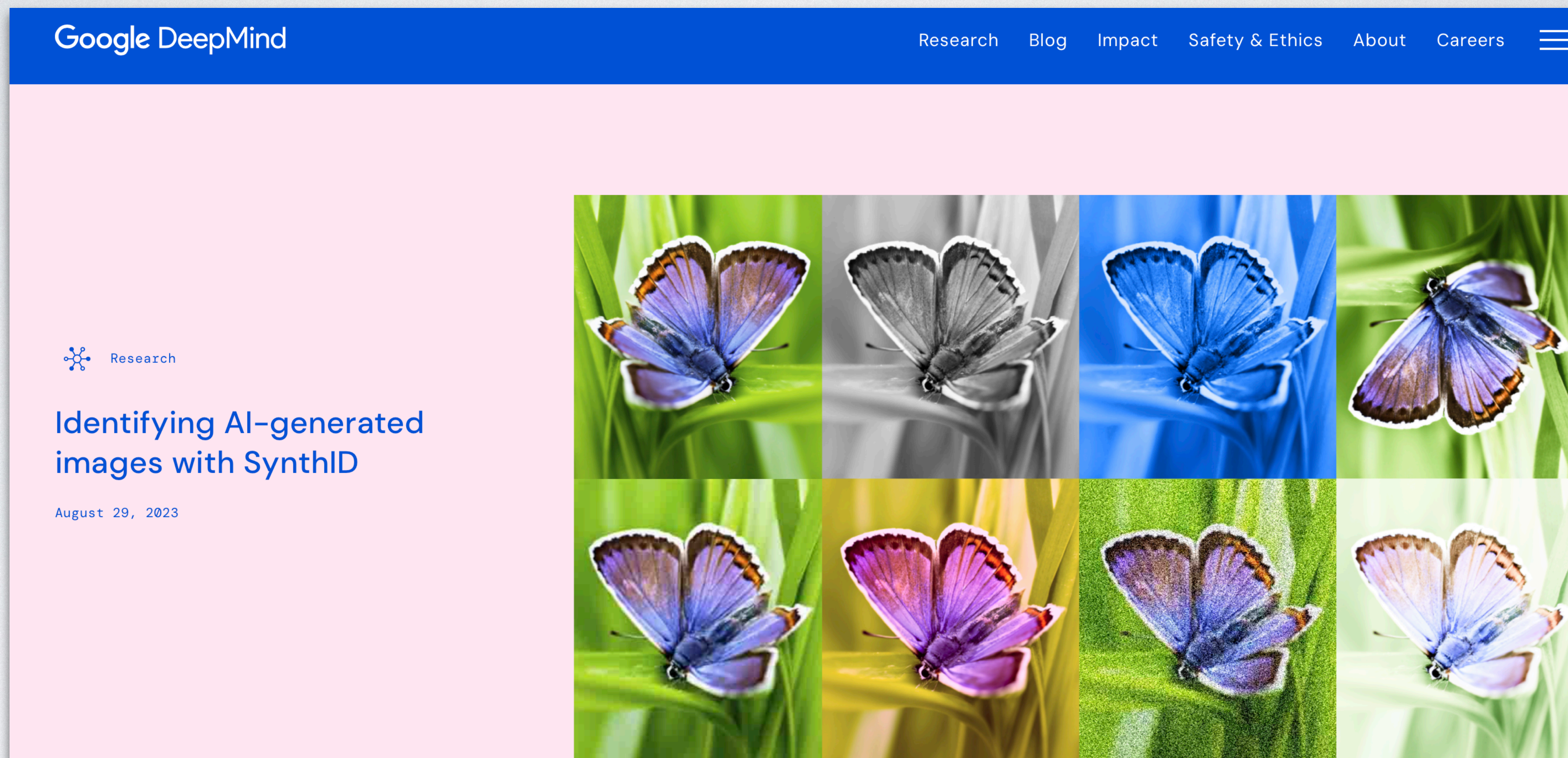
TRW Paper

Only  
non-adaptive  
Attackers





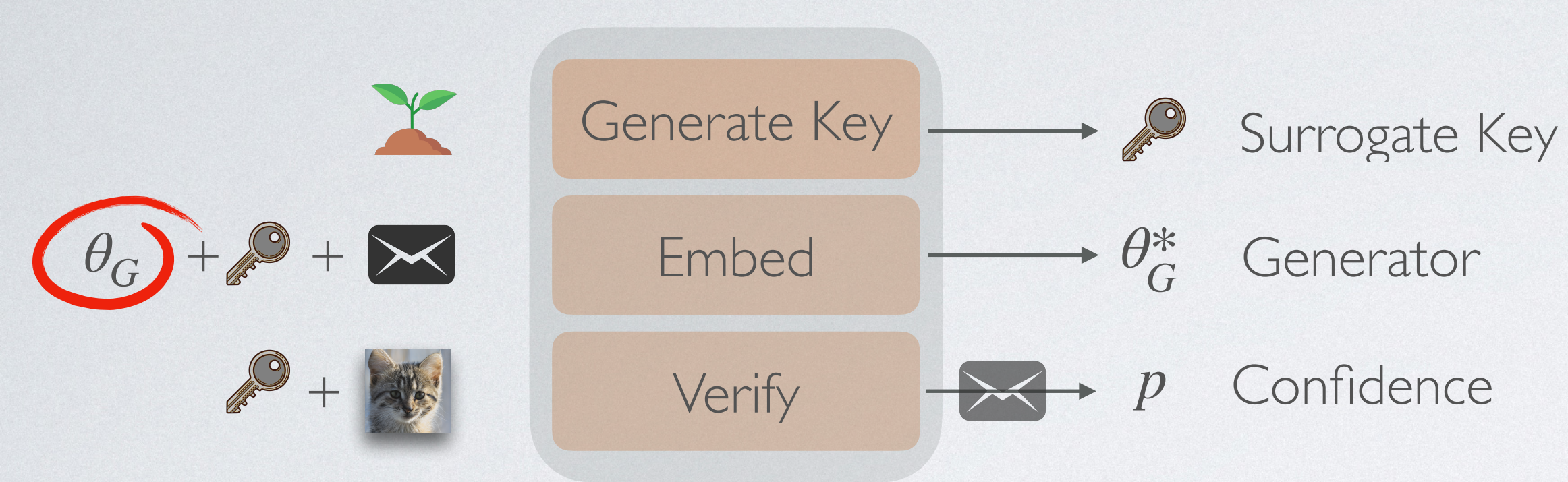
# Testing Robustness (SynthID)



Google SynthID, August 29th



# Threat Model



Watermarking  
Method



Attacker's Goals:

- (1) Evade detection ( $p \geq 0.01$ )
- (2) Preserve image quality



No access to the  
secret key



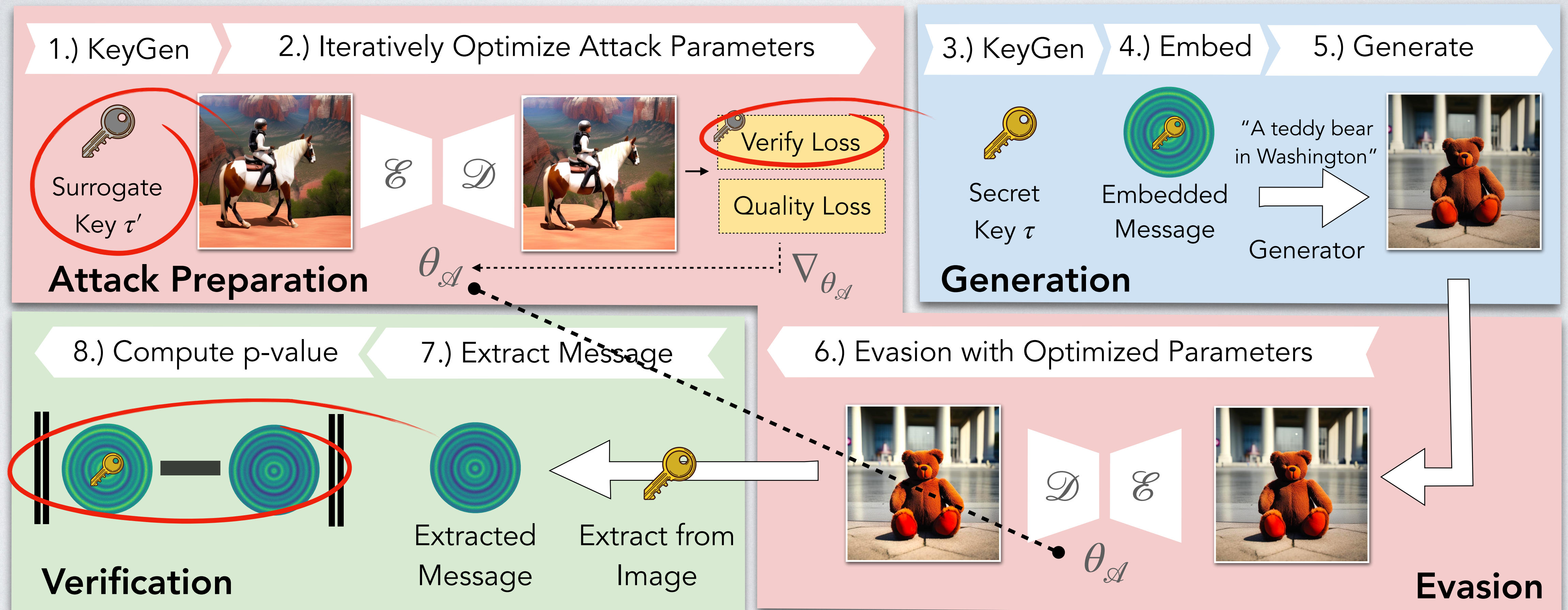
Watermarked  
Generator



Surrogate  
Generator  
(Less Capable)



# Instantiating Adaptive Attacks





# Optimization Goal

Best attack

Not necessarily differentiable!

Surrogate key

For any Key-message pair

Image after attack

Perceptual Similarity before and after ↓

$$\max_{\theta_{\mathcal{A}}} \mathbb{E}_{\substack{\tau' \leftarrow \text{KEYGEN}(\theta_G) \\ m \in \mathcal{M}}} [\text{VERIFY}(\mathcal{A}(G_W), \tau', m) + Q(\mathcal{A}(G_W), G_W)]$$

$m$  

$\tau'$  

$G_W$  

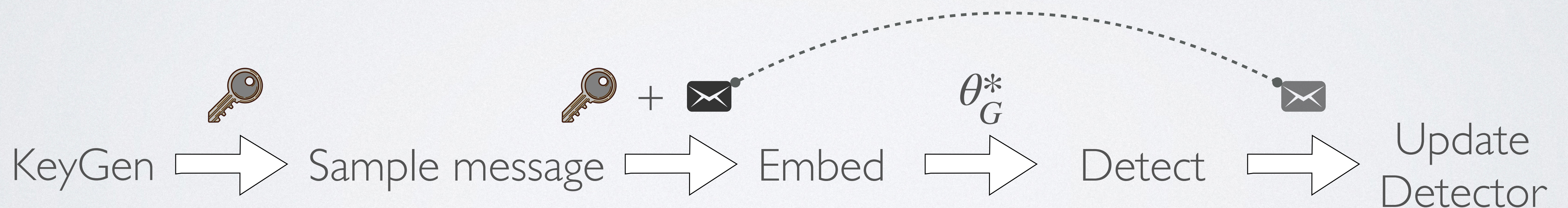


# Optimization Goal

$$\max_{\theta_{\mathcal{A}}} \mathbb{E}_{\substack{\tau' \leftarrow \text{KEYGEN}(\theta_G) \\ m \in \mathcal{M}}} [\text{VERIFY}(\mathcal{A}(G_W), \tau', m) + Q(\mathcal{A}(G_W), G_W)]$$

Simple solution to make VERIFY differentiable ..

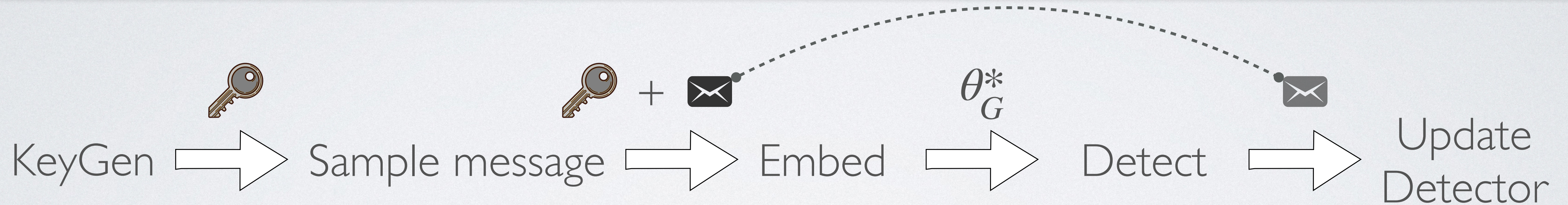
Train a deep classifier to extract the message





# Optimization Goal

$$\max_{\theta_A} \mathbb{E}_{\substack{\tau' \leftarrow \text{KEYGEN}(\theta_G) \\ m \in \mathcal{M}}} [\text{VERIFY}(\mathcal{A}(G_W), \tau', m) + Q(\mathcal{A}(G_W), G_W)]$$



**Observation:** In existing methods, KeyGen is not (sufficiently) randomized

Using a single surrogate key gives us a good approximation already



# Instantiating Adaptive Attacks

## Algorithm 2 Adversarial Noising

**Require:** surrogate  $\hat{\theta}_G$ , budget  $\epsilon$ , image  $x$

- 1:  $\theta_A \leftarrow 0$  ▷ adversarial perturbation
- 2:  $\theta_D \leftarrow \text{GKEYGEN}(\hat{\theta}_G)$
- 3:  $m \leftarrow \text{EXTRACT}(x; \theta_D)$
- 4: **for**  $j \leftarrow 1$  **to**  $N$  **do**
- 5:      $m' \leftarrow \text{EXTRACT}(x + \theta_A, \theta_D)$
- 6:      $g_{\theta_A} \leftarrow -\nabla_{\theta_A} \|m - m'\|_1$
- 7:      $\theta_A \leftarrow P_\epsilon(\theta_A - \text{Adam}(\theta_A, g_{\theta_A}))$
- return**  $x + \theta_A$

## Algorithm 3 Adversarial Compression

**Require:** surrogate  $\hat{\theta}_G$ , strength  $\alpha$ , image  $x$

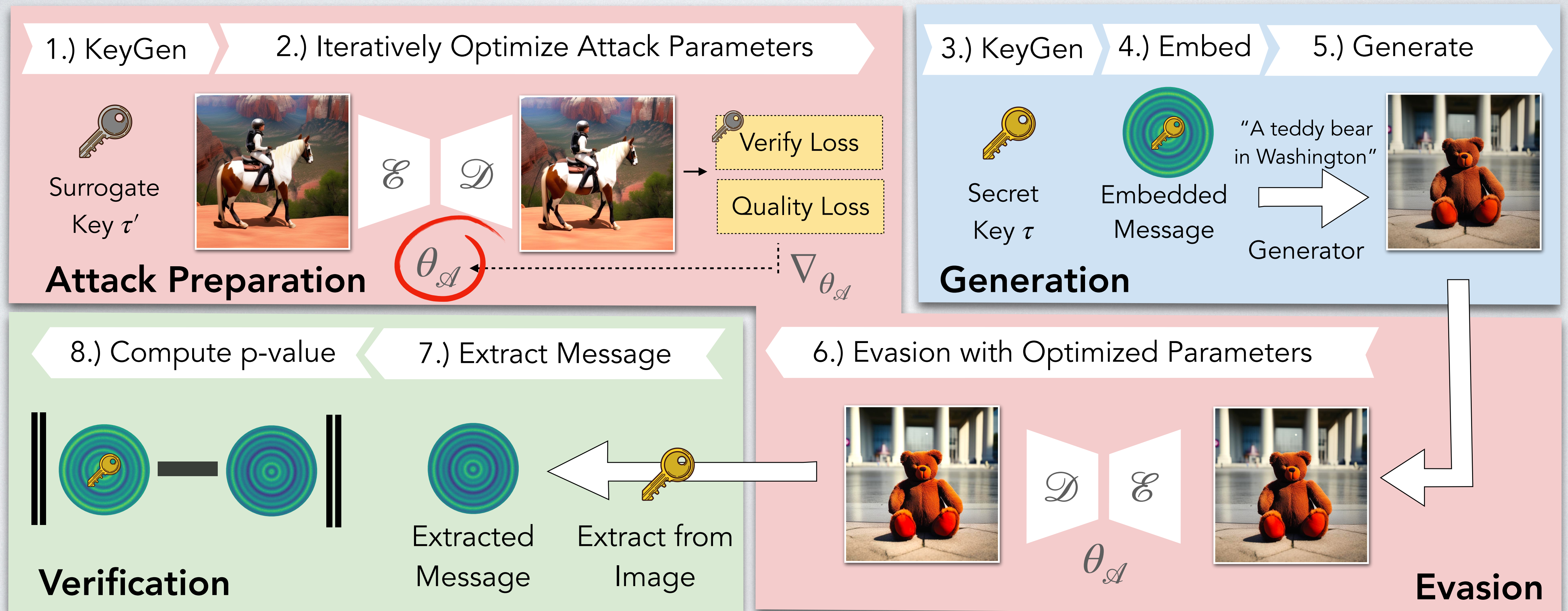
- 1:  $\theta_A \leftarrow [\theta_E, \theta_D]$  ▷ Compressor parameters
- 2:  $\theta_D \leftarrow \text{GKEYGEN}(\hat{\theta}_G)$  ▷ surrogate key
- 3: **for**  $j \leftarrow 1$  **to**  $N$  **do**
- 4:      $m \sim \mathcal{M}$
- 5:      $\hat{\theta}_G^* \leftarrow \text{EMBED}(\hat{\theta}_G, \theta_D, m)$
- 6:      $x \leftarrow \text{GENERATE}(\hat{\theta}_G^*)$
- 7:      $x' \leftarrow \mathcal{D}(\mathcal{E}(x; \theta_A))$  ▷ compression
- 8:      $m' \leftarrow \text{EXTRACT}(x', \theta_D)$
- 9:      $g_{\theta_A} \leftarrow \nabla_\delta(\mathcal{L}_{\text{LIPS}}(x', x) - \alpha \|m - m'\|_1)$
- 10:      $\theta_A \leftarrow \theta_A - \text{Adam}(\theta_A, g_{\theta_A})$
- return**  $\mathcal{D}(\mathcal{E}(x; \theta_A))$

Less than 1 million parameters

Around 80 million parameters



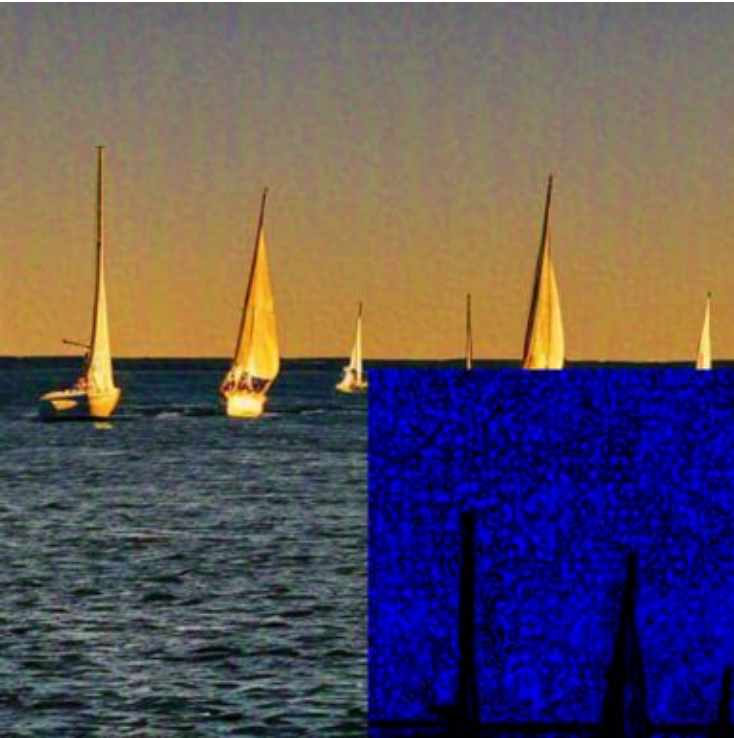




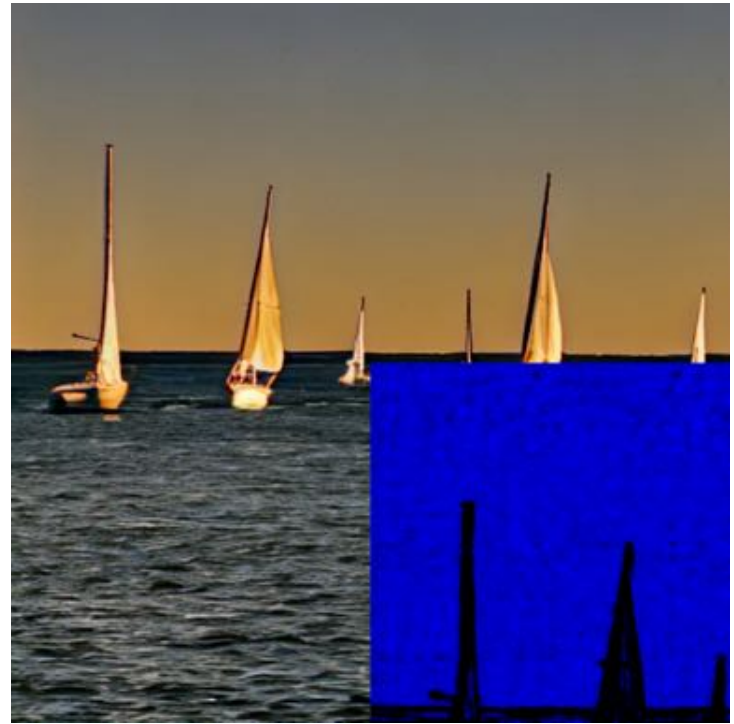




# Instantiating Adaptive Attacks



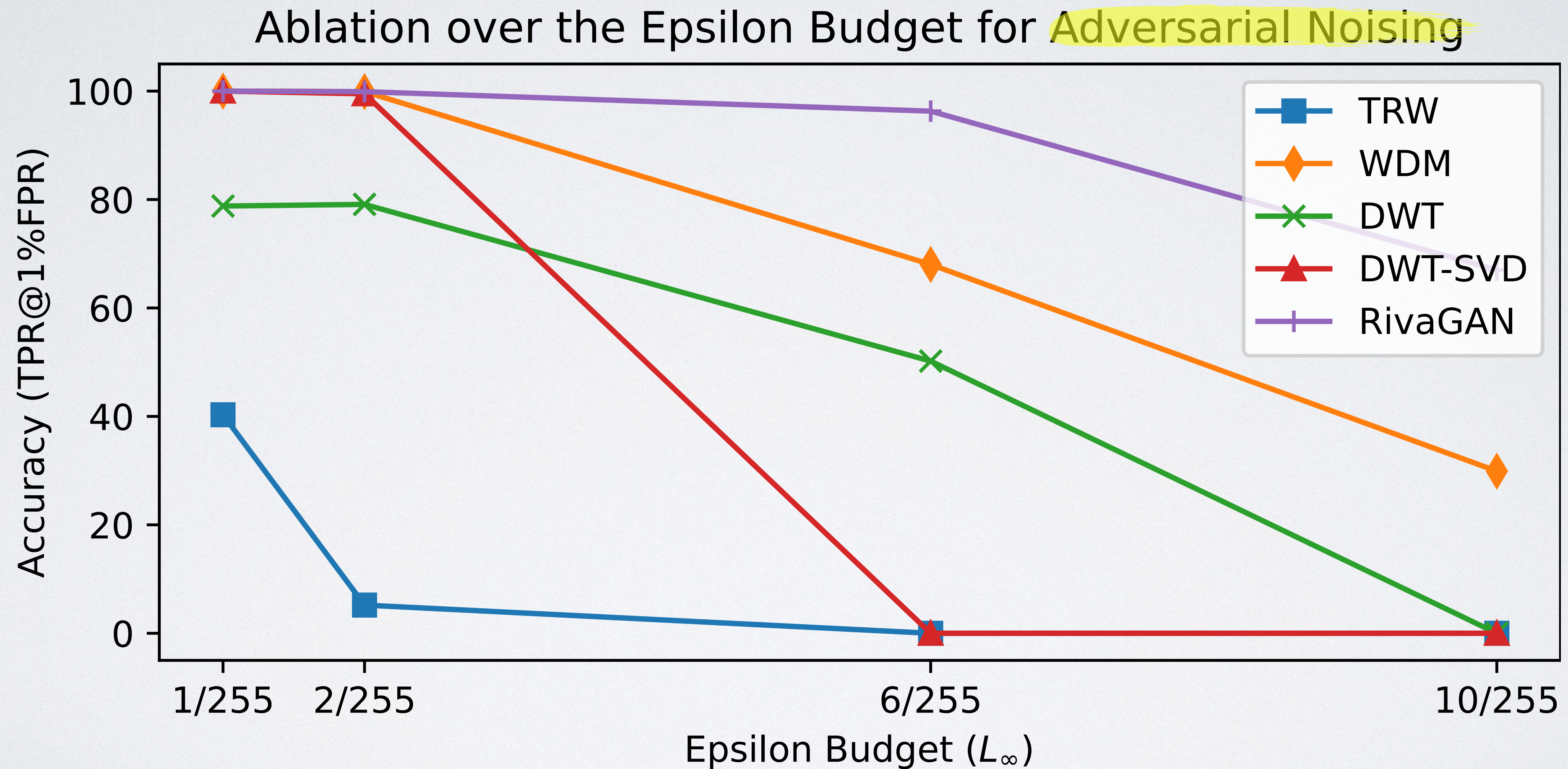


# Instantiating Adaptive Attacks

		TRW	WDM	DWT	DWT-SVD	RivaGAN
No Watermark	Adversarial Noise	 $\epsilon = 2/255, p = 0.09$	 $\epsilon = 8/255, p = 0.13$	 $\epsilon = 6/255, p = 0.18$	 $\epsilon = 4/255, p = 0.29$	 $\epsilon = 8/255, p = 0.05$
	Adversarial Compression	 $r = 1, p = 0.69$	 $r = 1, p = 0.79$	 $r = 1, p = 0.30$	 $r = 1, p = 1.00$	 $r = 3, p = 0.89$



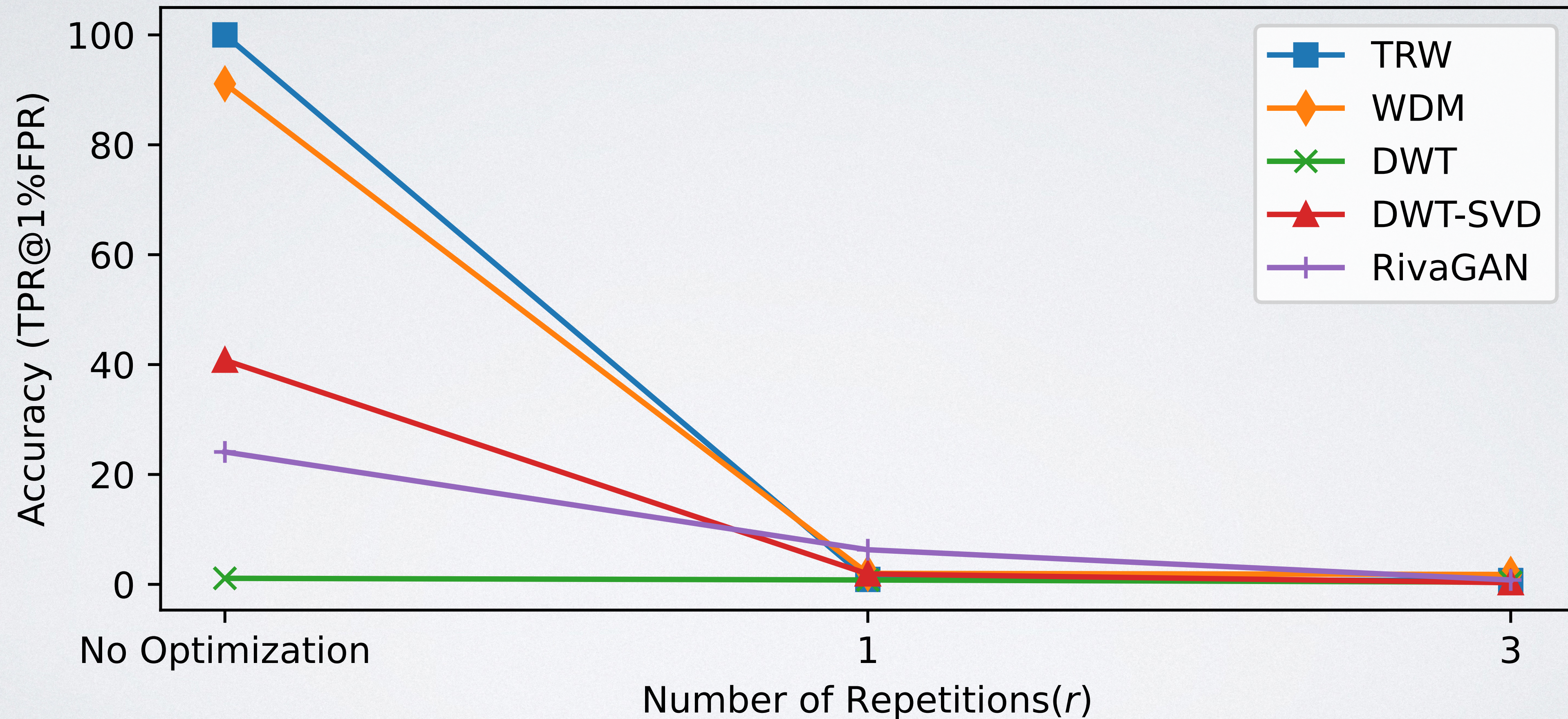
# Instantiating Adaptive Attacks





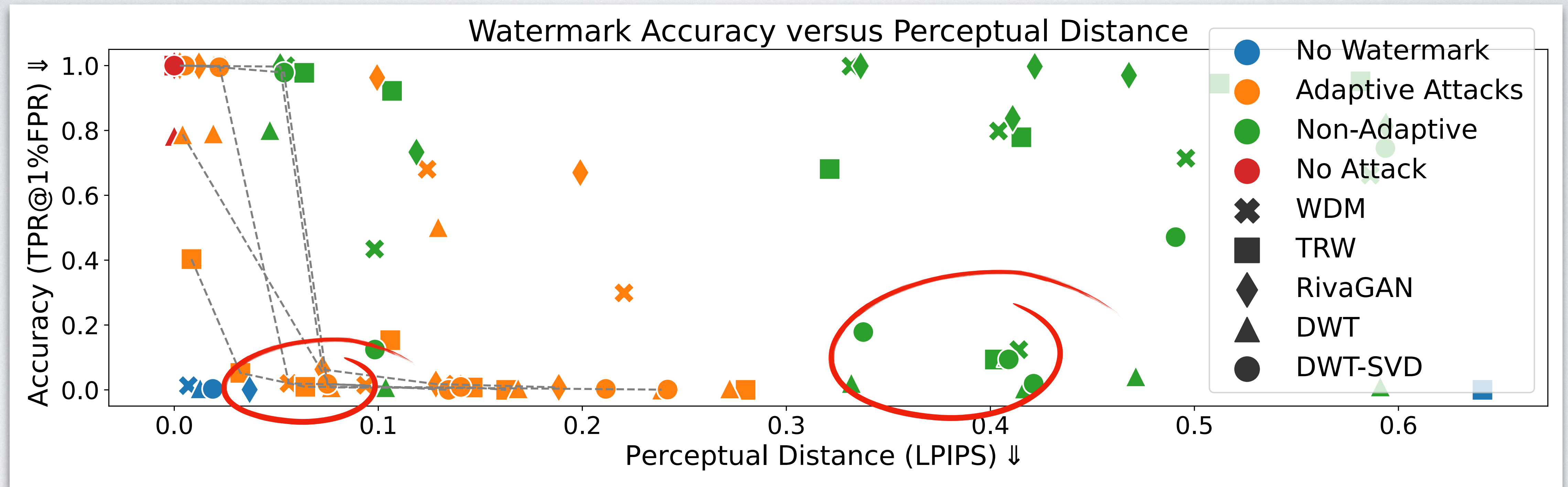
# Instantiating Adaptive Attacks

Ablation over the Repetitions for Adversarial Compression





# Comparison to Non-Adaptive Attacks



Adaptive Attacks

Non-adaptive Attacks



# Visual Inspection

**TRW: “Cars are parked on the street near an old building”**



P-value = 0.28



P-value = 1.77e-09



P-value = 0.52

$$\epsilon = 2/255, L_{\infty}\text{-norm}$$



ed on the street near an old building”





# Can we defend against Adaptive Attackers?

Best parameters

Not necessarily differentiable!

Surrogate key

$$\max_{\theta_A} \mathbb{E}_{\substack{\tau' \leftarrow \text{KEYGEN}(\theta_G) \\ m \in \mathcal{M}}} [\text{VERIFY}(\mathcal{A}(G_W), \tau', m) + Q(\mathcal{A}(G_W), G_W)]$$

For any Key-message pair

Image after attack

Image quality before and after



# Can we defend against Adaptive Attackers?

## Problem

TRW is not easily fixable against these adaptive attacks

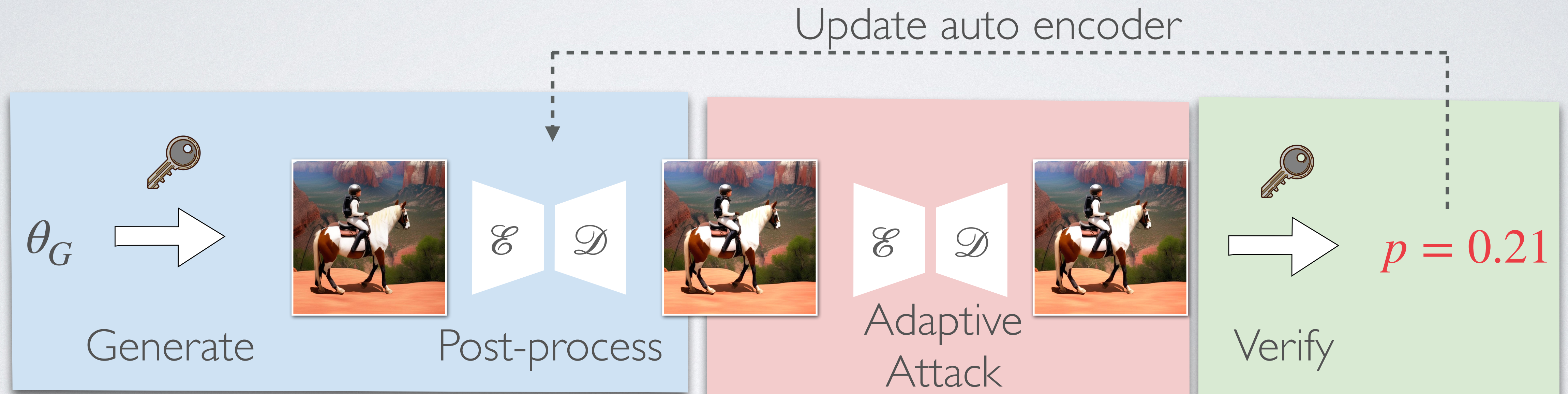
## Possible Solutions

*Learnable* watermarks, in which we train encoder-decoder pairs  
But how can we design them?



# Can we defend against Adaptive Attackers?

Idea I: **Post-processing**

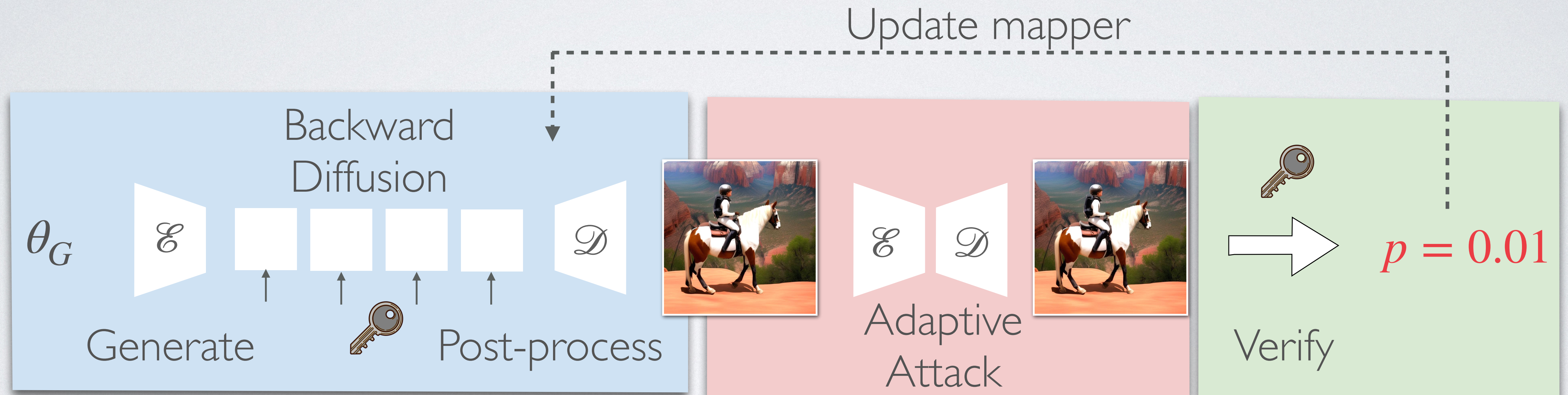


**Problem: Is the space of possible defense strategies large enough?  
There may not be an (efficient) solution!**



# Can we defend against Adaptive Attackers?

## Idea 2: Distributional Shift

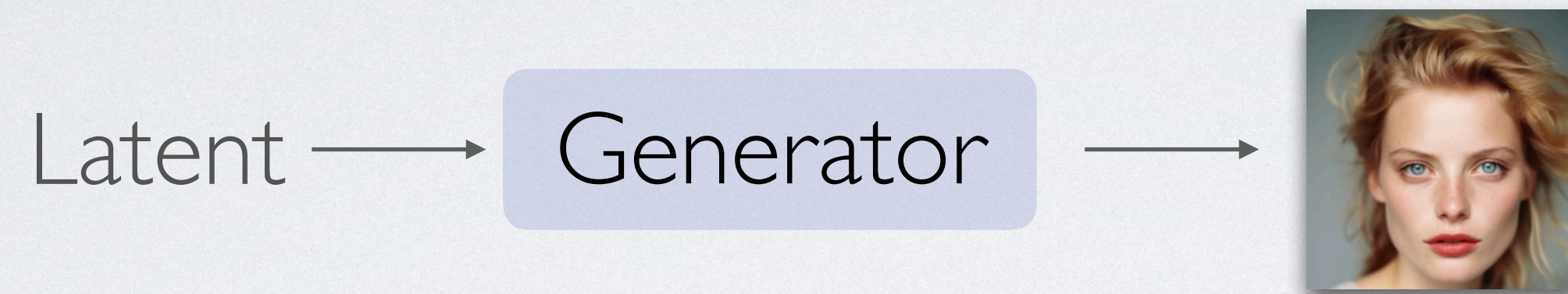


**Problem: Is there an (efficient) solution?**



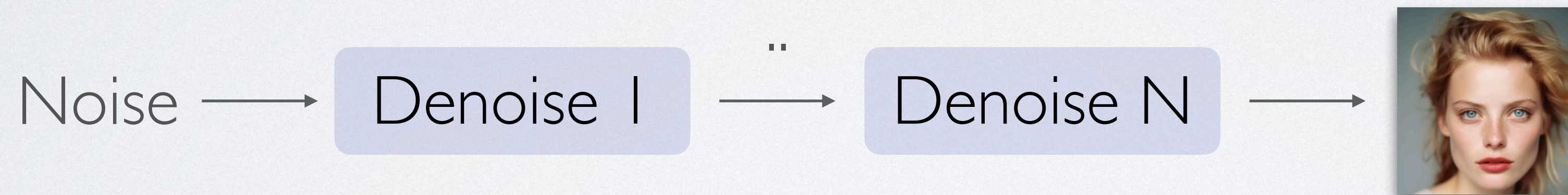
# Challenges of Learnable Watermarking

## 1.) One-shot agents (e.g., GANs)



- ✗ All gradients observable
- ✗ Alignment through pivot
- ✗ Continuous & high-entropy

## 2.) Iterative Optimization (e.g., Stable Diffusion)



## 3.) Discrete Iterative Optimization (e.g., Language Models)





# Discussion

How scalable are these attacks?

Will open-source model contain robust watermarks?

Certiably robust watermarking?

Extension to Language/Speech?

Ethical considerations

Limitations



# The Paper contains more Information



## LEVERAGING OPTIMIZATION FOR ADAPTIVE ATTACKS ON IMAGE WATERMARKS

Nils Lukas, Abdulrahman Diaa, Lucas Fenaux, Florian Kerschbaum  
University of Waterloo, Canada

{nilukas, abdulrahman.diaa, lucas.fenaux, florian.kerschbaum}@uwaterloo.ca

### ABSTRACT

Untrustworthy users can misuse image generators to synthesize high-quality deepfakes and engage in online spam or disinformation campaigns. Watermarking deters misuse by marking generated content with a hidden message, enabling its detection using a secret watermarking key. A core security property of watermarking is robustness, which states that an attacker can only evade detection by substantially degrading image quality. Assessing robustness requires designing an adaptive attack for the specific watermarking algorithm. A challenge when evaluating watermarking algorithms and their (adaptive) attacks is to determine whether an adaptive attack is optimal, i.e., it is the best possible attack. We solve this problem by defining an objective function and then approach adaptive attacks as an optimization problem. The core idea of our adaptive attacks is to replicate secret watermarking keys locally by creating *surrogate keys* that are differentiable and can be used to optimize the attack's parameters. We demonstrate for Stable Diffusion models that such an attacker can break all five surveyed watermarking methods at negligible degradation in image quality. These findings emphasize the need for more rigorous robustness testing against adaptive, learnable attackers.

**Keywords** Watermarking, Stable Diffusion, Robustness, Adaptive Attacks

### 1 Introduction

Deepfakes are images synthesized using deep image generators that can be difficult to distinguish from real images. While deepfakes can serve many beneficial purposes if used ethically, for example, in medical imaging [Akrout et al., 2023] or education [Peres et al., 2023] they also have the potential to be *misused* and erode trust in digital media. Deepfakes have already been used in disinformation campaigns [Boneh et al., 2019] and social engineering attacks [Mirsky and Lee, 2021], highlighting the need for methods that control the misuse of deep image generators.

Watermarking offers a solution to controlling misuse by embedding hidden messages into all generated images that are later detectable using a secret watermarking key. Images that are detected as deepfakes can be flagged by social media platforms or news agencies, which can mitigate potential harm [Grinbaum and Adomaitis, 2022]. Providers of large image generators such as Google have announced the deployment of their own watermarking methods [Gowal and Kohli, 2023] to enable the detection of deepfakes and promote the ethical use of their models.

A core security property of watermarking is *robustness*, which states that an attacker can evade detection only by substantially degrading the image's quality. While several watermarking methods have been proposed for image generators [Wen et al., 2023, Zhao et al., 2023, Fernandez et al., 2023], none of them are certifiably robust [Bansal et al., 2022] and instead, robustness is tested empirically using a limited set of known attacks. Claimed security properties of previous watermarking methods have been broken by novel attacks [Lukas et al., 2022], and no comprehensive method exists to validate robustness, which causes difficulty in trusting the deployment of watermarking in practice.

We propose testing the robustness of watermarking by defining robustness using objective function and approaching adaptive attacks as an optimization problem. Adaptive attacks are specific to the watermarking algorithm used by the defender but have no access to the secret watermarking key. Knowledge of the watermarking algorithm enables the attacker to consider a range of *surrogate keys* similar to the defender's key. This is also a challenge for optimization since the attacker only has imperfect information about the optimization problem. Adaptive attackers had previously

### Leveraging Optimization for Adaptive Attacks on Image Watermarks

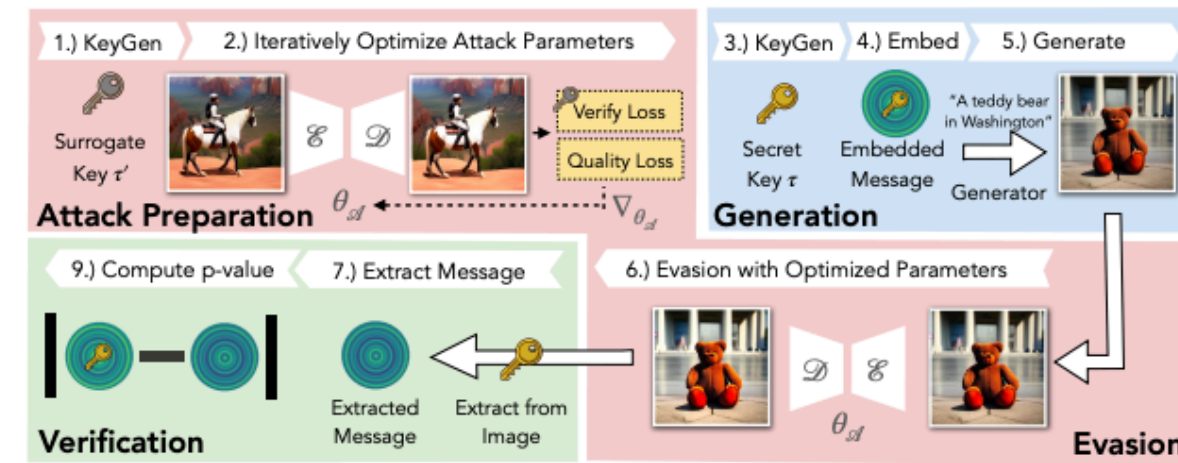


Figure 1: An overview of our adaptive attack pipeline. The attacker prepares their attack by generating a surrogate key and leveraging optimization to find optimal attack parameters  $\theta_A$  (illustrated here as an encoder  $\mathcal{E}$  and decoder  $\mathcal{D}$ ) for any message. Then, the attacker generates watermarked images and applies a modification using their optimized attack to evade detection. The attacker succeeds if the verification procedure cannot detect the watermark in high-quality images.

been shown to break the robustness of watermarking for image classifiers [Lukas et al., 2022], but attacks had to be handcrafted against each watermarking method. Finding attack parameters through an optimization process can be challenging when the watermarking method is not easily optimizable, for instance, when it is not differentiable. Our attacks leverage optimization by approximating watermark verification through a differentiable process. We show that adaptive, *learnable* attackers, whose parameters can be optimized efficiently, can evade watermark detection for 1 billion parameter Stable Diffusion models at a negligible degradation in image quality.

### 2 Background

**Latent Diffusion Models (LDMs)** are state-of-the-art generative models for image synthesis [Rombach et al., 2022]. Compared to Diffusion Models [Sohl-Dickstein et al., 2015], LDMs operate in a latent space using fixed, pre-trained autoencoder consisting of an image encoder  $\mathcal{E}$  and a decoder  $\mathcal{D}$ . LDMs use a forward and reverse diffusion process across  $T$  steps. In the forward pass, real data point  $x_0$  is encoded into a latent point  $z_0 = \mathcal{E}(x_0)$  and is progressively corrupted into noise via Gaussian perturbations. Specifically,

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1 - \beta_t}z_{t-1}, \beta_t\mathbf{I}), \quad t \in \{0, 1, \dots, T-1\},$$

where  $\beta_t$  is the scheduled variance. In the reverse process, a neural network  $f_\theta$  guides the denoising, taking  $z_t$  and time-step  $t$  as inputs to predict  $z_{t-1}$  as  $f_\theta(z_t, t)$ . The model is trained to minimize the mean squared error between the predicted and actual  $z_{t-1}$ . The outcome is a latent  $\hat{z}_0$  resembling  $z_0$  that can be decoded into  $\hat{x}_0 = \mathcal{D}(\hat{z}_0)$ . Synthesis in LDMs can be conditioned with textual prompts.

## Extended Evaluation

Watermarking embeds a hidden signal into a medium, such as images, using a secret watermarking key that is later extractable using the same secret key. In the context of deep learning, watermarking can be characterized by the medium used by the defender to verify the presence of the hidden signal. White-box and black-box watermarking methods assume access to the model's parameters or query access via an API respectively, and have been used primarily for Intellectual Property protection [Uchida et al., 2017].

*No-box* watermarking [Lukas and Kerschbaum, 2023] assumes a more restrictive setting where the defender only knows the generated content but does not know the query used to generate the image. This type of watermarking has been used to control misuse by having the ability to detect if an image generated by the provider image generator [Gowal and Kohli, 2023]. Given a set of images  $\{x_i\}_{i=1}^n$ , a *No-box* watermarking method defines a function  $f$  that takes these images as

## Detailed Algorithmic Descriptions

## Discussion & Ethics

### Leveraging Optimization for Adaptive Attacks on Image Watermarks

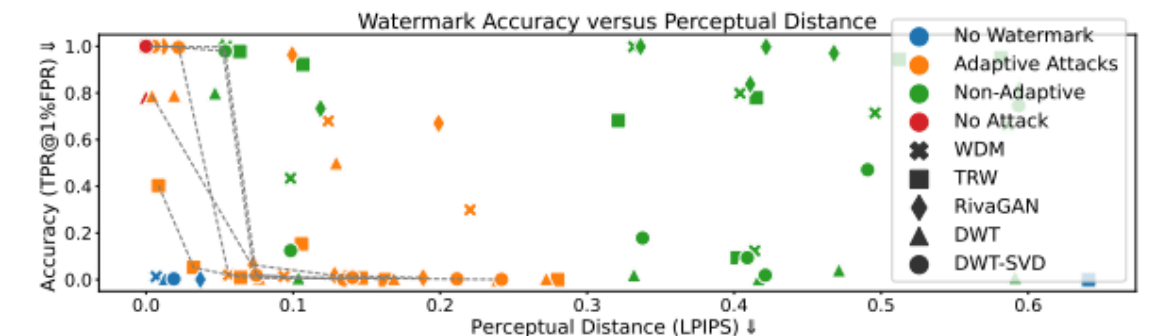


Figure 2: The effectiveness of our attacks against all watermarks. We highlight the Pareto front for each watermarking method by dashed lines and indicate adaptive/non-adaptive attacks by colors.

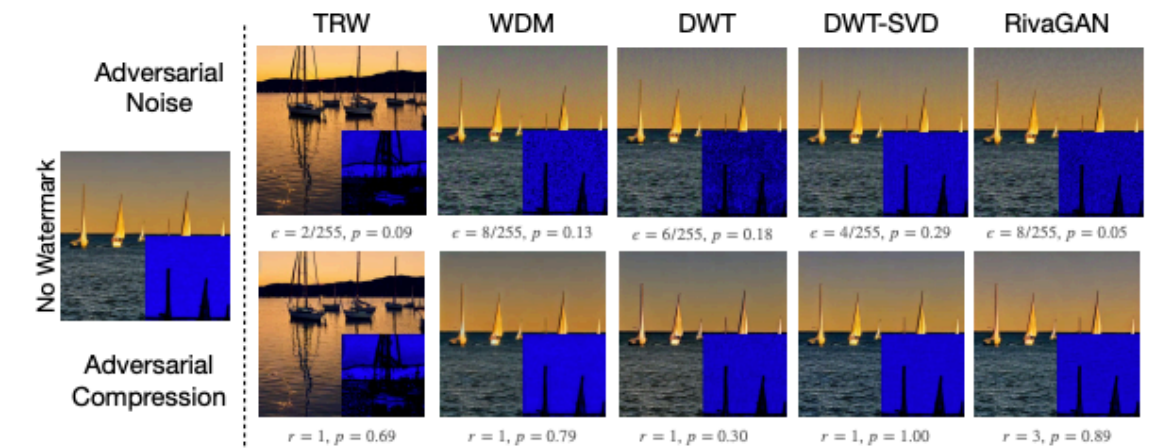


Figure 3: A visual analysis of two adaptive attacks. The left image shows the unwatermarked output, including a high-contrast cutout of the top left corner of the image to visualize noise artifacts. On the right are images after evasion with adversarial noising (top) and adversarial compression (bottom).

### 5.2 Image Quality after an Attack

Figure 3 shows the perceptual quality after using our adaptive attacks. We show a cutout of the top left image patch with high contrasts on the bottom right to visualize noise artifacts potentially introduced by our attacks. We observe that, unlike adversarial noising, the compression attack introduces no new visible artifacts. Appendix A.3 displays more visualizations on the perceptual impact of our attacks on the image quality.

	TRW		WDM		DWT		DWT-SVD		RivaGAN	
	FID	CLIP	FID	CLIP	FID	CLIP	FID	CLIP	FID	CLIP
No WM	23.32	31.76	23.48	31.77	23.48	31.77	23.48	31.77	23.48	31.77
WM	24.19	31.78	23.43	31.72	23.16	32.11	23.10	32.15	22.96	31.84
A-Noise	23.67	32.15	N/A	N/A	23.55	32.46	22.89	32.50	N/A	N/A
A-Comp	24.36	31.87	23.27	32.01	23.16	32.17	23.06	31.92	23.25	31.86

Table 2: Quality metrics before and after watermark evasion. FID $\downarrow$  represents the Fréchet Inception Distance, and CLIP $\uparrow$  represents the CLIP score, computed on 5k images from MS-COCO-2017. N/A means the attack could not evade watermark detection, and we do not report quality measures.

Table 2 shows the FID and CLIP score of the watermarked images and the images after using adversarial noising and adversarial compression. We calculate the quality using the best attack configuration from Figure 2 when the detection



# The Paper contains more Information



## PTW: Pivotal Tuning Watermarking for Pre-Trained Image Generators

Nils Lukas  
University of Waterloo

Florian Kerschbaum  
University of Waterloo

### Abstract

Deepfakes refer to content synthesized using deep generators, which, when *misused*, have the potential to erode trust in digital media. Synthesizing high-quality deepfakes requires access to large and complex generators only a few entities can train and provide. The threat is malicious users that exploit access to the provided model and generate harmful deepfakes without risking detection. Watermarking makes deepfakes detectable by embedding an identifiable code into the generator that is later extractable from its generated images. We propose Pivotal Tuning Watermarking (PTW), a method for watermarking pre-trained generators (i) three orders of magnitude faster than watermarking from scratch and (ii) without the need for any training data. We improve existing watermarking methods and scale to generators  $4\times$  larger than related work. PTW can embed longer codes than existing methods while better preserving the generator's image quality. We propose rigorous, game-based definitions for robustness and undetectability and our study reveals that watermarking is not robust against an adaptive white-box attacker who has control over the generator's parameters. We propose an adaptive attack that can successfully remove any watermarking with access to only 200 non-watermarked images. Our work challenges the trustworthiness of watermarking for deepfake detection when the parameters of a generator are available.

### 1 Introduction

Deepfakes, a term used to describe synthetic media generated using deep image generators have received widespread attention in recent years. While deepfakes offer many beneficial use cases, for example in scientific research [9, 48] or education [16, 39, 47], they have also raised ethical concerns because of their potential to be *misused* which can lead to an erosion of trust in digital media. Deepfakes have been scrutinized for their use in disinformation campaigns [2, 23], impersonation attacks [15, 35] or when used to create non-consensual media of an individual violating their privacy [10, 20]. These threats highlight the need to control the misuse of deepfakes.

While some deepfakes can be created using traditional computer graphics, using deep learning methods such as the Generative Adversarial Network (GAN) [19] can reduce the time and effort needed to create deepfakes. However, training GANs requires a significant investment in terms of computational resources [26] and data preparation, including collection, organization, and cleaning. These costs make training image generators a prohibitive endeavor for many. As a consequence, generators are often trained by one *provider* and made available to many users through Machine-Learning-as-a-Service [6]. The provider wants to disclose their model responsibly and deter *model misuse*, which is the unethical use of their model to generate harmful or misleading content [36].

**Problem.** Consider a provider who wants to make their image generator publicly accessible under a contractual usage agreement that serves to prevent misuse of the model. The threat is a user who breaks this agreement and uses the generator to synthesize and distribute harmful deepfakes without detection. To mitigate this threat in practice, companies such as OpenAI have deployed invasive prevention measures by providing only monitored access to their models through a black-box API. Users that synthesize deepfakes are detectable when they break the usage agreement if the provider matches the deepfake with their database. This helps deter misuse of the model, but it can also lead to a lack of transparency and limit researchers and individuals from using their technology [12, 50]. For example, query monitoring which is used in practice by companies such as OpenAI raises privacy concerns as it involves collecting and potentially storing sensitive information about the user's queries. A better solution would be to implement methods that deter model misuse without the need for query monitoring.

A potential solution is to rely on deepfake detection methods [7, 13, 17, 24, 25, 30, 40, 56]. The idea guiding such *passive* methods is to exploit artifacts in the synthetic images that separate fake and real content. While these detectors protect well against some deepfakes it has been demonstrated that

<sup>§</sup>To appear at USENIX Security 2023.

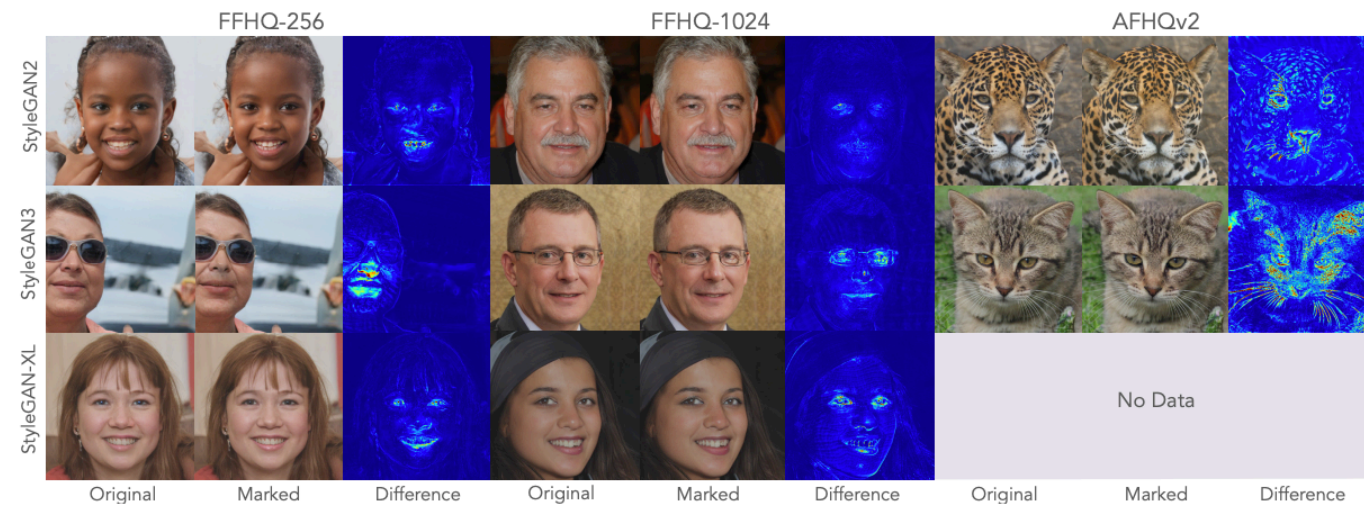


Figure 5: Images synthesized using our watermarked generators on different datasets and model architectures. We show the image synthesized by the generator (i) before and (ii) after watermarking, and (iii) the difference between the watermarked and non-watermarked images. StyleGAN-XL does not provide a pre-trained model checkpoint for AFHQv2.

**Capacity.** We measure the capacity of a watermark in bits by the difference in the expected number of correctly extracted bits from watermarked and non-watermarked images. The expected rate of correctly extracted bits equals 0.5 for non-watermarked images assuming messages are sampled uniformly at random. Let  $m \in \{0, 1\}^n$  be a message,  $\tau$  the secret watermarking key, and  $\theta$  are the parameters of a generator. The capacity of the generator is computed as follows.

$$C_\theta = n \cdot \mathbb{E}_{z \sim \mathcal{Z}} [\text{VERIFY}(\text{EXTRACT}(G(z; \theta), \tau)) - 0.5] \quad (5)$$

It is straightforward to achieve a high capacity by overwriting a significant portion of the host image. However, this approach also decreases the visual image quality, which can be measured and visualized as the capacity/utility trade-off.

**Decision Threshold.** We consider a watermark to be *removed*, if we can reject the null hypothesis  $H_0$  with a  $p$ -value less than 0.05. The null hypothesis states that  $k$  matching bits were extracted from the synthetic images by random chance. Quantitatively the probability of this event is calculated as  $\Pr(X = k | H_0) = \sum_{i=0}^k \binom{n}{i} 0.5^n$ . For a watermark with  $n = 40$  bits, we need to reject at least 26 bits correctly, meaning that we verify the presence of a watermark by correctly extracting  $C_\theta \geq 26$  in bits.

### 5.3 Runtime Analysis

To calculate the speed-up of PTW over existing watermarking methods [60, 61], we compare it with training non-watermarked generators from scratch. This comparison is fair, as watermarking is not expected to decrease a generator's training time. We estimate the total runtimes in GPU hours using the suggested hyperparameters in the relevant GAN papers [28, 51] on an A100 GPU.

Model	StyleGAN2	StyleGAN3	StyleGAN-XL
FFHQ-256	158h	482h	552h
FFHQ-512	384h	662h	1285h
FFHQ-1024	929h	1161h	1456h

Table 2: GPU hours required for training generators without watermarking from scratch on FFHQ [27] on 8xA100 GPUs.

Table 2 shows the estimated training runtimes from scratch for each generator on FFHQ [28] at varying pixel resolutions. For instance, training a StyleGAN2 model on FFHQ at a resolution of 256 pixels requires 158 GPU hours. With PTW, watermarking a pre-trained generator on FFHQ requires only about 0.5 GPU hours which is a three-orders of magnitude improvement for high-resolution generators. Our approach also requires training the watermarking decoder (see Algorithm 4), which is a one-time upfront cost of about 2 GPU hours.

### 5.4 Capacity/utility Trade-off

This section summarizes our results on the capacity/utility trade-off on various datasets, model architectures, and in comparison to existing modified watermarks: Yu1, and Yu2. Visual inspection. Figure 5 shows images synthesized by our watermarked generators on all three surveyed datasets. The columns show the original image synthesized before watermarking, the image synthesized after watermarking and their differences in the form of a heatmap. Heavily modified regions are highlighted in yellow and red. In both versions of the facial image datasets, we observe that our watermark focuses on levels located on the edge of the generated person's most prominently on the eyes. Upon closer inspection, the net

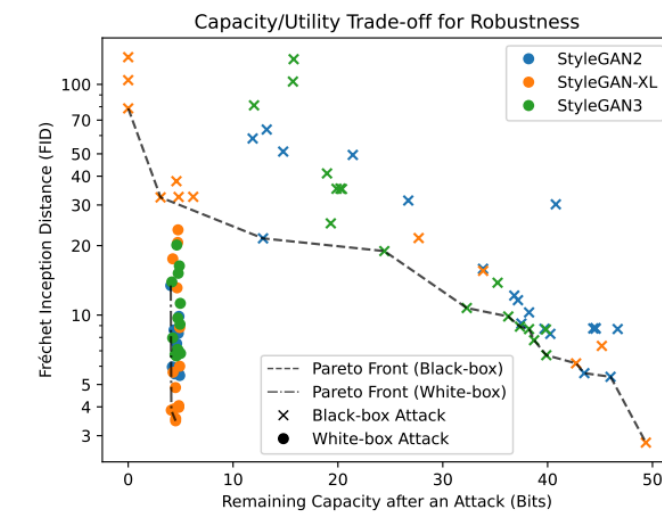


Figure 8: This Figure shows the robustness of our watermark against all surveyed attacks. We highlight black-box and white-box attacks that are members of the Pareto front.

off a black-box attacker can achieve using these attacks. For example, a black-box attacker can reduce the capacity by 10 bits from 50 to 40, but in doing so reduces the FID by over 6 points. Our super-resolution attack is on the Pareto front but cannot remove the watermark. Removal is only possible when the FID drops to 30, at which point the image quality has been compromised. Table 3 summarizes the best-performing black-box attacks for the three evaluated generator architectures. Each attack has a single parameter that we ablate over using grid search. We refer to Appendix A for a detailed description of all attacks and parameters we used in this ablation. Table 3 lists those data points that either remove the watermark ( $C_\theta < 5$ ) or, if the watermark cannot be removed, the data point with the lowest FID. None of the black-box attacks, including our super-resolution attack, are successful in removing the watermark while preserving the generator's utility.

### 5.6.2 White-box Attacks

**Overwriting.** Table 3 shows that overwriting can remove watermarks but deteriorates the generator's image quality, measured using FID, by approximately 3 points for StyleGAN2 and 6 points for StyleGAN-XL. Such a deterioration in FID likely prevents attacks in practice because low-quality deepfakes are more easily detectable. Our overwriting attack also implicitly assumes knowledge of the defender's watermarking method which may not be the case in practice. Overwriting could cause a greater decline in FID if the attacker's and defender's watermarking methods differ.

**Reverse Pivotal Tuning.** Our Reverse Pivotal Tuning (RPT) attack is substantially more effective than the overwriting attack as it preserves the FID of the generator to a greater extent. We found that an attacker with access to 200

	StyleGAN2		StyleGAN-XL		StyleGAN3	
Attacks	$C_\theta$	FID	$C_\theta$	FID	$C_\theta$	FID
<b>Black-box Attacks</b>						
Crop	39.73	8.72	42.71	6.18	38.23	8.69
Blur	38.82	36.84	12.12	10.32	35.12	11.73
JPEG	42.12	8.70	38.43	9.12	38.23	9.33
Noise	40.26	8.29	45.17	7.35	32.29	10.73
Quantize	37.17	11.60	43.27	5.61	39.72	8.71
SR	32.86	11.51	34.52	11.62	30.12	11.34
<b>White-box Attacks</b>						
Overwrite	4.78	8.34	4.91	8.83	4.73	9.71
RPT <sub>200</sub>	4.91	5.47	4.52	3.52	4.59	6.65
RPT <sub>100</sub>	4.44	5.56	4.21	3.90	4.47	6.75
RPT <sub>50</sub>	4.38	8.07	4.38	15.32	4.16	14.47

Table 3: The capacity and FID of all surveyed attacks. We ablate over multiple parameters for each attack and this table shows the points with the best (i.e., lowest) FID. RPT<sub>R</sub> stands for the Reverse Pivotal Tuning attack using  $R$  real samples.

real, non-watermarked images is capable of removing any watermark without causing a noticeable deterioration in FID. This means that with access to less than 0.3% of the training dataset, a white-box adversary can remove any watermark. In the case of StyleGAN-XL, using 200 images leads to a decrease in FID of less than one point (from 2.67 to 3.52).

**Ablation Study for RPT.** Figure 7c shows an ablation study over the amount of real, non-watermarked training data required by an attacker to remove a watermark. We measured these curves as follows: We randomly sample a set of  $R$  real images and run the RPT attack encoded by Algorithm 5 with gradually increasing weight  $\lambda_{\text{LIPS}}$  on the LIPS loss until the watermark is removed. Then we compute the FID on  $K = 50,000$  images. In all experiments, the watermark is eventually removed but access to more data has a significant impact on the FID that is retained in the generator after the attack. For StyleGAN2, we find that 80 images ( $\approx 0.1\%$  of the training data) are sufficient to remove the watermark at less than 0.3 points of deterioration in FID, which represents a visually imperceptible quality degradation. Our results demonstrate that an adaptive attacker with access to the generator's parameters can remove any watermark using only a small number of clean, non-watermarked images and can pose a threat to the trustworthiness of watermarking.

### 6 Discussion

This section discusses the limitations of watermarking and our study, the extension of our work to other image generators, and ethical considerations from releasing our attacks.

**Non-Cooperative Providers.** Our study demonstrates that watermarking for image generators can be robust under cer-

# Robustness with more attacks



# How Reliable is Watermarking for Generative Machine Learning?

Source code: <https://github.com/dnn-security/gan-watermark>



Nils Lukas



UNIVERSITY OF  
**WATERLOO**



**CrySP**  
Cryptography, Security, and Privacy  
Research Group



Source Code



USENIX'23



# Sources

- [1] Karras, Tero, et al. "Alias-free generative adversarial networks." *Advances in Neural Information Processing Systems* 34 (2021): 852-863.
- [2] "How a fake image of a Pentagon explosion shared on Twitter caused a real dip on Wall Street", Luke Hurst, <https://www.euronews.com/next/2023/05/23/fake-news-about-an-explosion-at-the-pentagon-spreads-on-verified-accounts-on-twitter>, Accessed June 25 2023
- [3] "The viral AI-generated image showing an explosion near the Pentagon is 'truly the tip of the iceberg of what's to come,' tech CEO says", Grace Dean, <https://www.businessinsider.com/ai-generated-images-deepfake-pentagon-explosion-tip-of-the-iceberg-2023-6>, Accessed June 25 2023
- [4] "Fake Pentagon explosion photo goes viral: How to spot an AI image", Mohammed Haddad, <https://www.aljazeera.com/news/2023/5/23/fake-pentagon-explosion-photo-goes-viral-how-to-spot-an-ai-image>, Accessed June 25 2023
- [5] "Terms of use", OpenAI, <https://openai.com/policies/terms-of-use>, Accessed June 25 2023
- [6] "GENERATIVE AI PROHIBITED USE POLICY", Google, <https://policies.google.com/terms/generative-ai/use-policy>, Accessed June 25 2023
- [7] Saharia, Chitwan, et al. "Photorealistic text-to-image diffusion models with deep language understanding." *Advances in Neural Information Processing Systems* 35 (2022): 36479-36494.
- [8] Kang, Minguk, et al. "Scaling up gans for text-to-image synthesis." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
- [9] <https://www.louisbouchard.ai/latent-diffusion-models/>
- [10] "StyleGAN3 Synthetic Image Detection", NVIDIA, <https://github.com/NVLabs/stylegan3-detector>, Accessed June 25 2023
- [11] Dong, Chengdong, Ajay Kumar, and Eryun Liu. "Think Twice Before Detecting GAN-generated Fake Images from their Spectral Domain Imprints." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.






# Sources

- [12] “Russian TV and radio stations hacked with fake Putin message”, <https://www.dw.com/en/russian-tv-and-radio-stations-hacked-with-fake-putin-message/a-65830291>
- [13] “Russia Says 'Fake' Putin Address Declaring Martial Law Was a 'Hack'”, Matthew Gault, <https://www.vice.com/en/article/ak33ge/russia-says-fake-putin-address-declaring-martial-law-was-a-hack>
- [14] “Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS”, EU Legislation, <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206>, Accessed June 26th
- [15] <https://apnews.com/article/deepfake-porn-celebrities-dalle-stable-diffusion-midjourney-ai-e7935e9922cda82fbcfble1a88d9443a>, accessed August 9th
- [16] <https://www.nbcnews.com/tech/internet/deepfakes-twitter-tiktok-stars-rcna87295>, accessed August 9th
- [17] <https://www.ic3.gov/Media/Y2023/PSA230605>, accessed August 9th
- [18] <https://www.reuters.com/technology/openai-google-others-pledge-watermark-ai-content-safety-white-house-2023-07-21/>, accessed August 9th






# Best Adaptive Attacks




**TRW: "Cars are parked on the street near an old building"**

		
P-value = 0.28	P-value = 1.77e-09	P-value = 0.52

**WDM: "A bench at the beach next to the sea"**

		
P-value = 0.13	P-value = 3.73-11	P-value = 0.08

**DWT: "A blue train on some train tracks about to go under a bridge"**

		
P-value = 0.30	P-value = 2.33-10	P-value = 0.57

**DWT-SVD: "A white horse standing on top of a dirt field."**

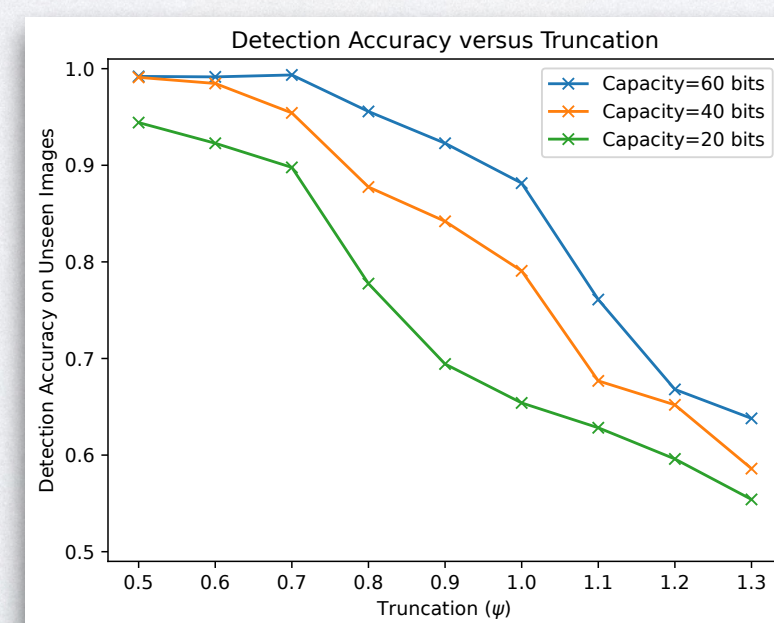
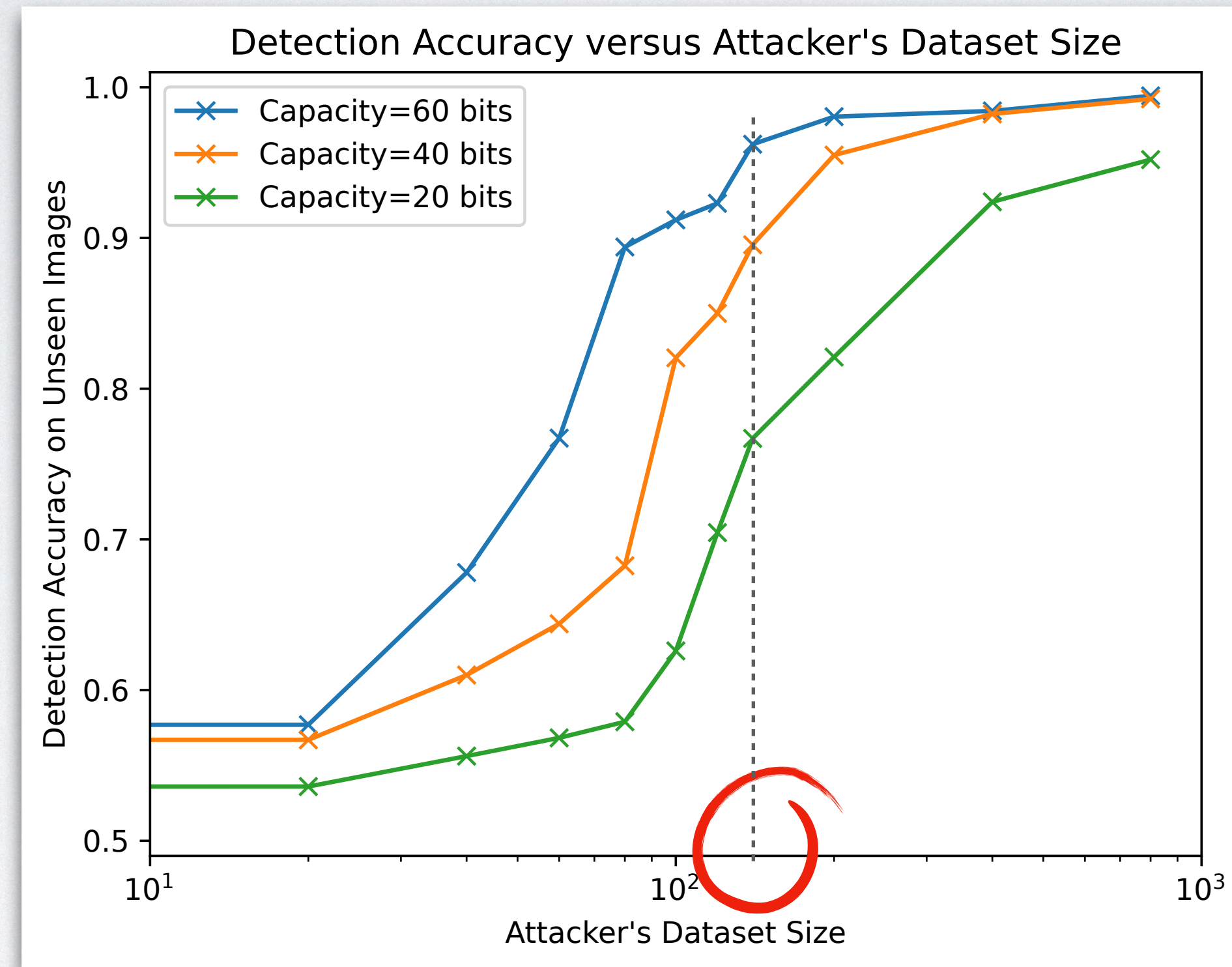
		
P-value = 0.30	P-value = 2.33-10	P-value = 0.05

**RivaGAN: "Donuts with frosting and glazed toppings sit on table next to coffee maker"**

		
P-value = 0.30	P-value = 2.33-10	P-value = 0.43



# Detectability

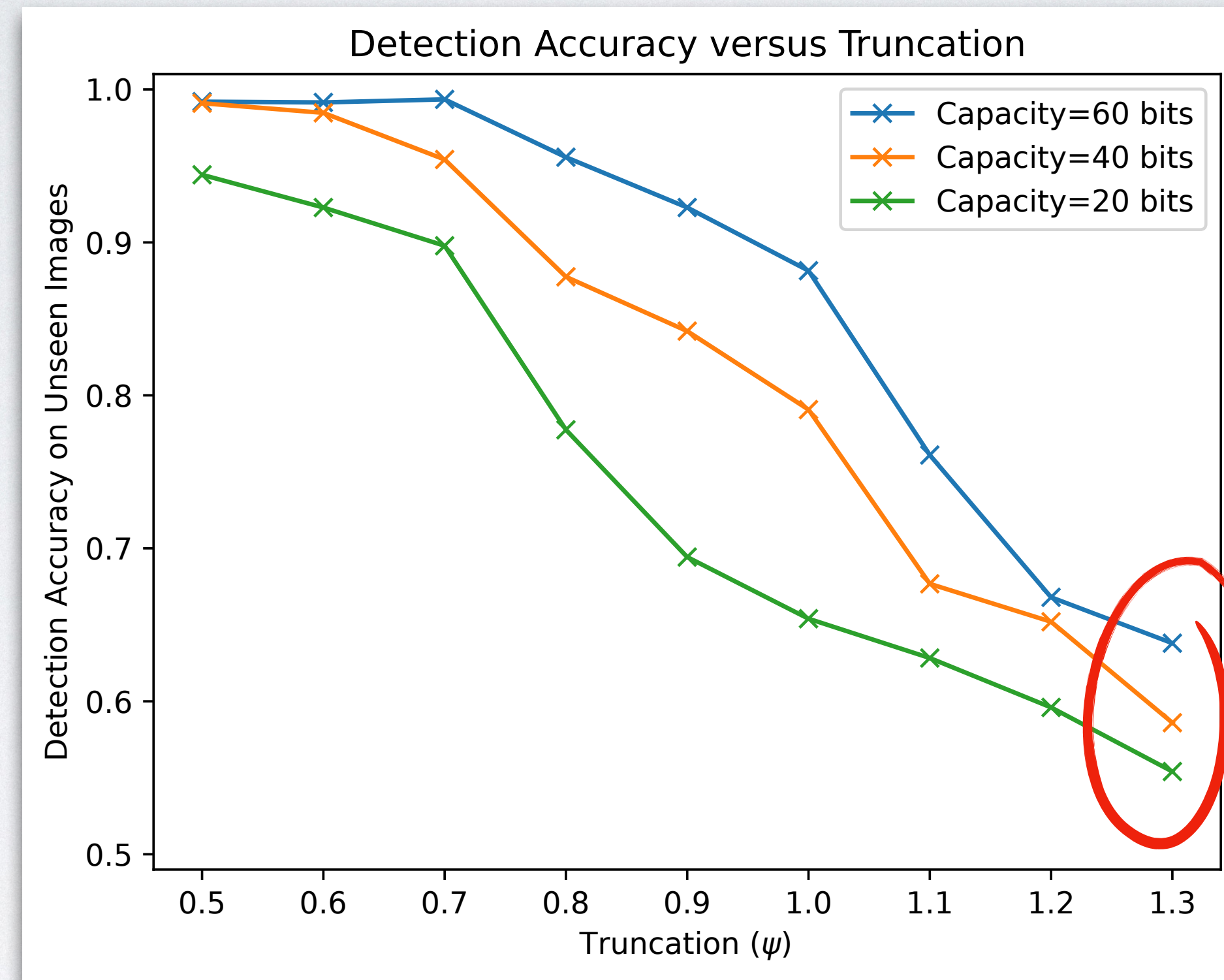




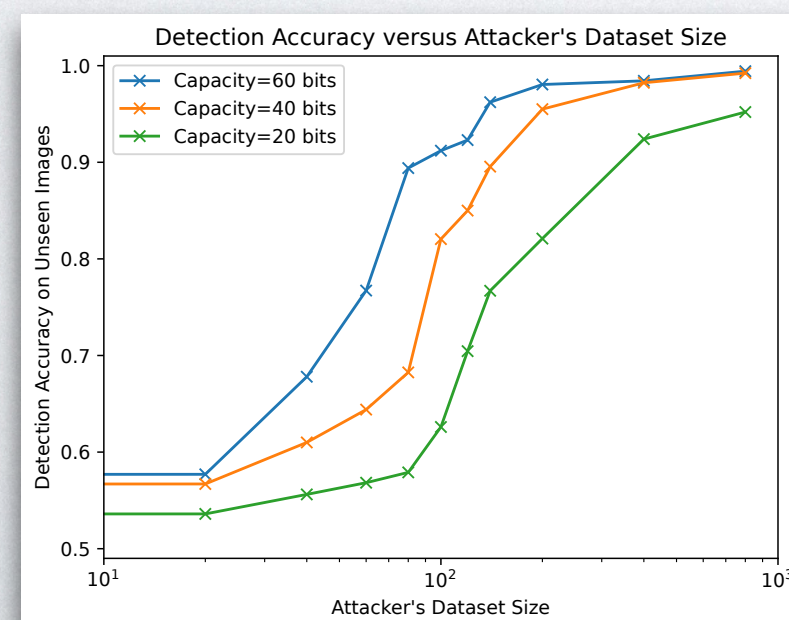
# Detectability



Low variation



High variation



Variation makes detectability more difficult for the adversary

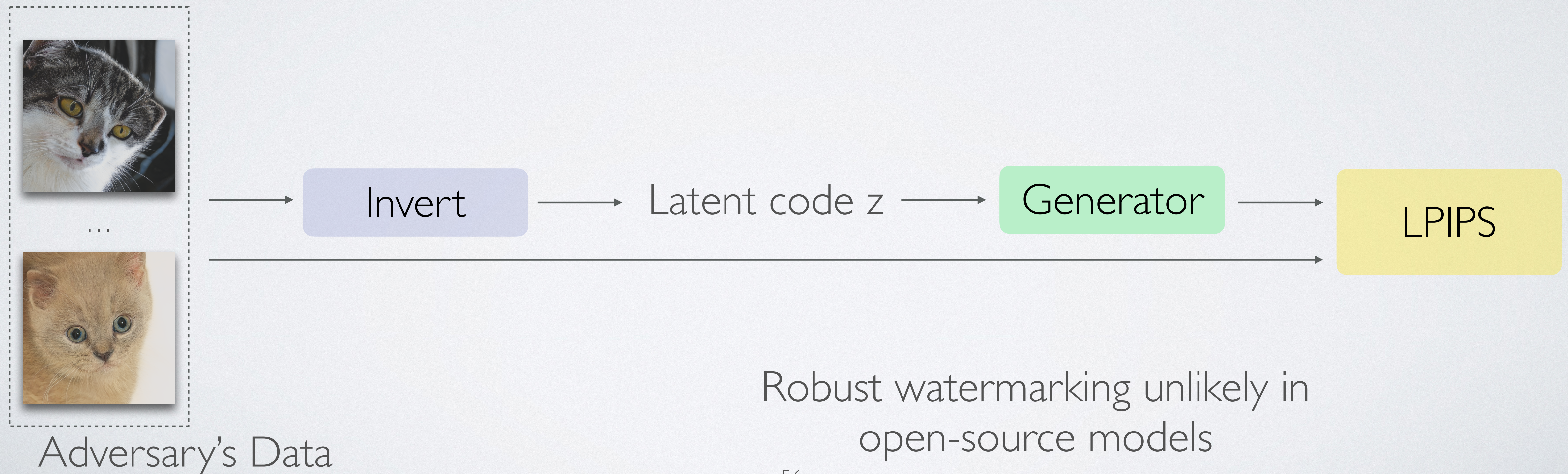


# Reverse Pivotal Tuning

White-box

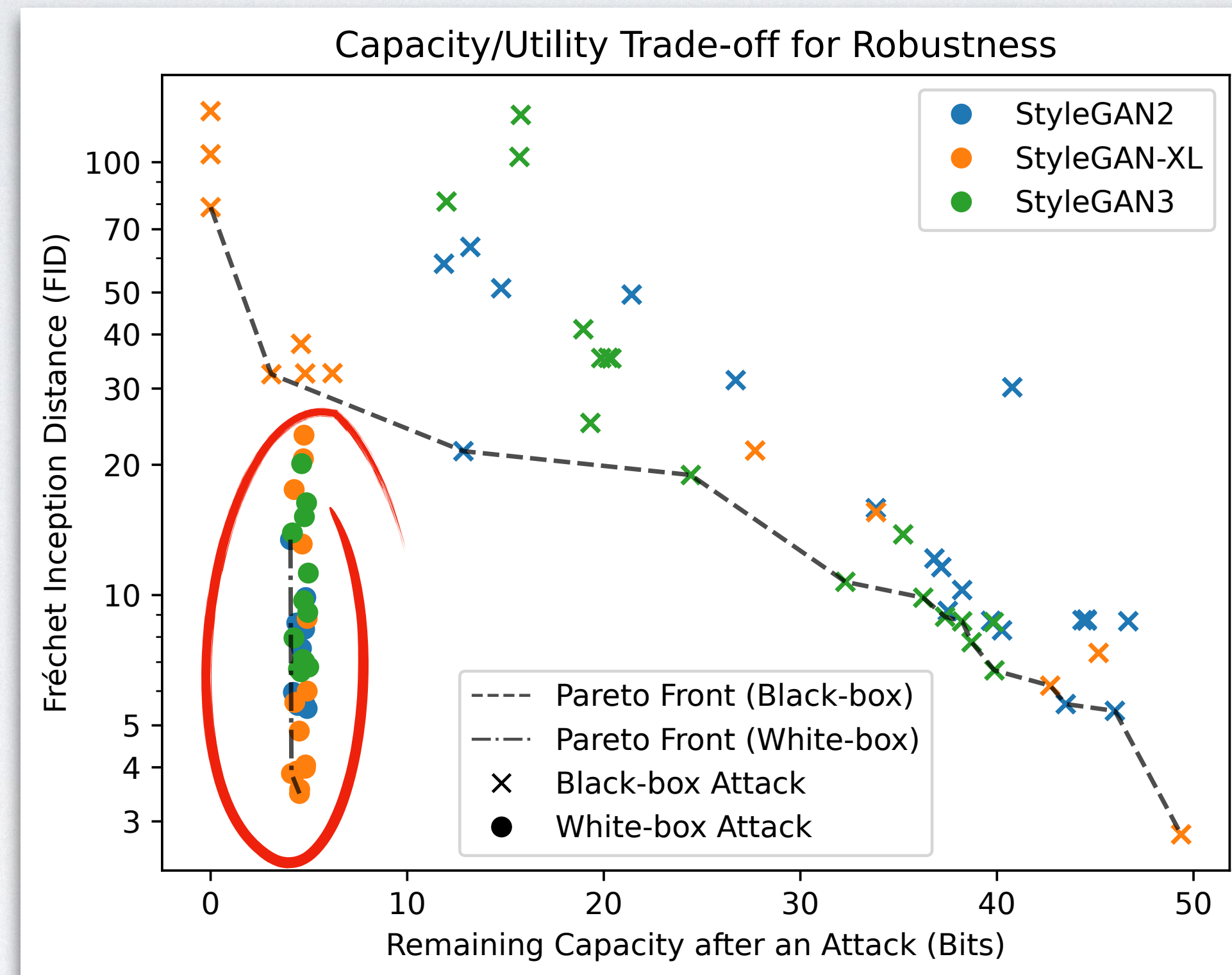
1) Invert real images into the generator's latent space

2) Regularize generator with Pivotal Tuning and LPIPS loss to synthesize real images





# Robustness



Black-box attacker  
cannot remove watermarks

White-box attacker  
can remove any watermark



# Summary of Results

	Black-box	White-box
Robustness	✓	✗
Detectability	Scales with output diversity	
Effectiveness	40 bits at less than 0.3 FID	
Scalability	No retraining, < 2 GPU hours (FFHQ-256)	

The first post-hoc **learnable watermark** for deep image generators