PTW: Pivotal Tuning Watermarking for Pre-Trained Image Generators

Nils Lukas

Florian Kerschbaum

CrySP
Cryptography, Security, and Privacy Research Group

UNIVERSITY OF WATERLOO

Talking Points

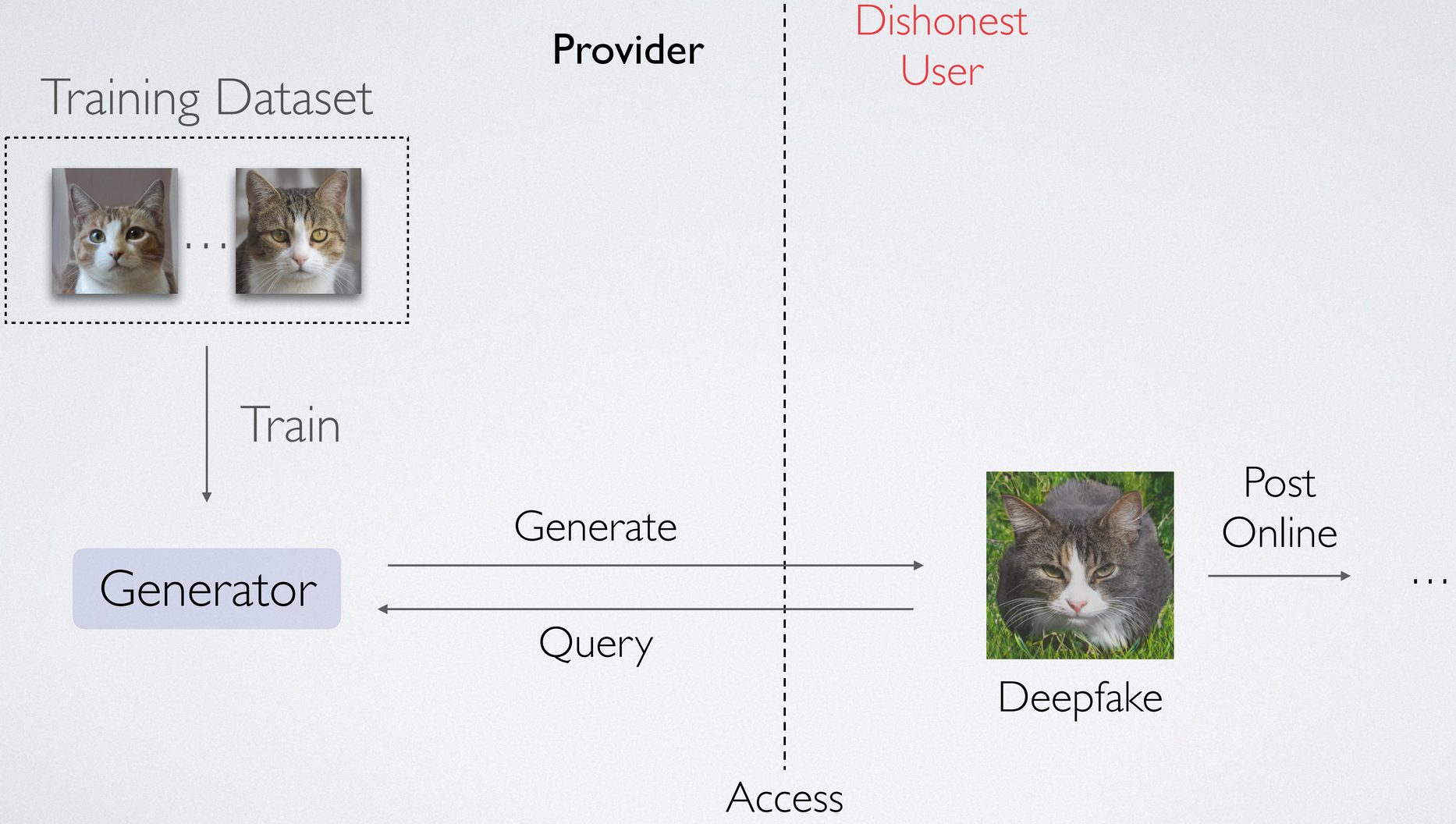
1) What is (intentional) Model Misuse ?

2) What can be done to prevent misuse?

3) Proposed solutions

4) Outlook

Model Misuse



Model Misuse

Deepfakes erode trust in digital media

Possible Attacks: Impersonation, disinformation,
non-consensual media of individuals (e.g., NSFW content)

Disinformation

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

May 23rd, Euronews [2]

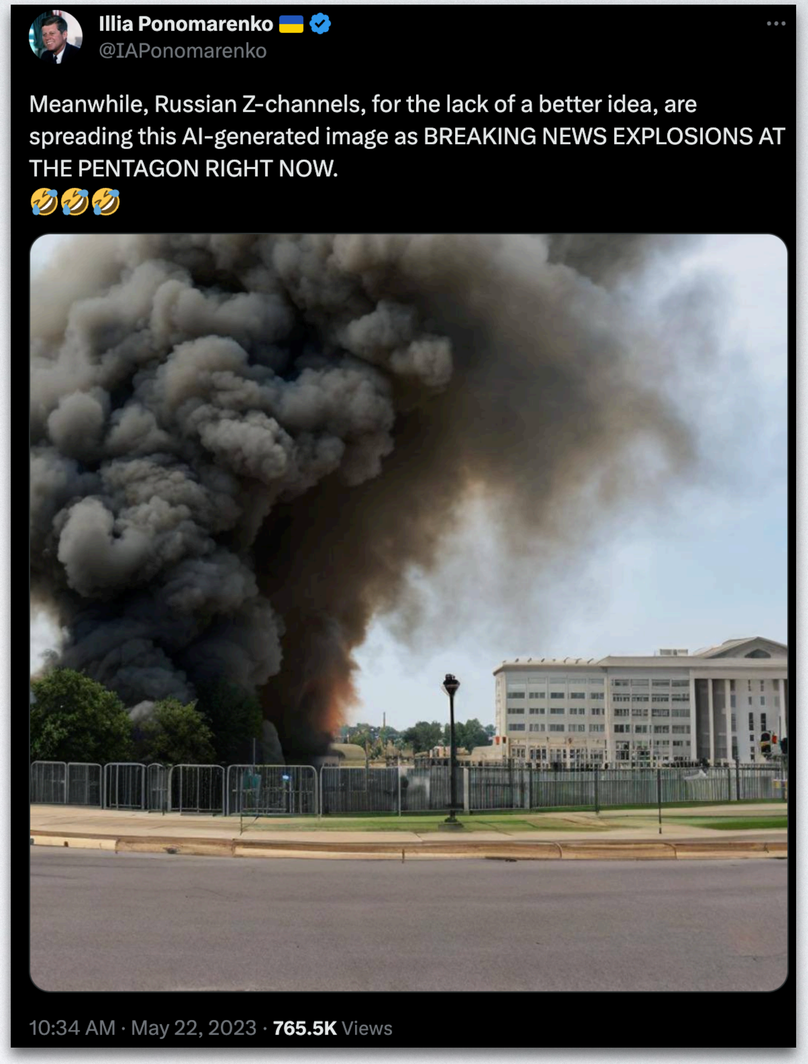
The viral AI-generated image showing an explosion near the Pentagon is 'truly the tip of the iceberg of what's to come,' tech CEO says

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

June 9th, Business Insider [3]

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

May 23rd, Aljazeera [4]



Disinformation

CONFLICTS | **RUSSIAN FEDERATION**

Russian TV and radio stations hacked with fake Putin message

06/05/2023

Hackers took over the broadcasts of several TV and radio stations in the regions bordering Ukraine. The Kremlin said authorities have regained control of the airwaves.

June 5th, Deutsche Welle [12]

Russia Says 'Fake' Putin Address Declaring Martial Law Was a 'Hack'

Putin appeared on TV and the radio and urged citizens to flee deeper into Russia.

By Matthew Gault

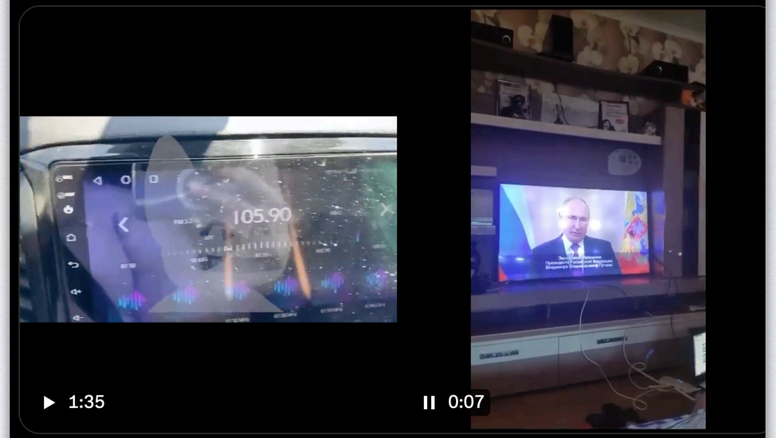
June 5th, Motherboard [13]

NEXTA
@nexta_tv

Fake "Putin's urgent appeal" was broadcast on radio and television in the border regions

With the voice of the bunker Fuhrer the hackers announced the martial law in Belgorod, Bryansk and Kursk regions, the general mobilization and evacuation deep into the Russian territory. The appeal was also shown on TV channels.

Peskov said that "Putin's emergency address shown on some TV networks is a hack."



8:45 AM · Jun 5, 2023 · **614.9K** Views

Draft Legislation

5.2.4. TRANSPARENCY OBLIGATIONS FOR CERTAIN AI SYSTEMS (TITLE IV)

Title IV concerns certain AI systems to take account of the specific risks of manipulation they pose. Transparency obligations will apply for systems that (i) interact with humans, (ii) are used to detect emotions or determine association with (social) categories based on biometric data, or (iii) generate or manipulate content ('deep fakes'). When persons interact with an AI system or their emotions or characteristics are recognised through automated means, people must be informed of that circumstance. If an AI system is used to generate or manipulate image, audio or video content that appreciably resembles authentic content, there should be an obligation to disclose that the content is generated through automated means, subject to exceptions for legitimate purposes (law enforcement, freedom of expression). This allows persons to make informed choices or step back from a given situation.

2021, EU Legislation Draft [14]

Controlling Misuse

(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 [5]

2. Generate and distribute content intended to misinform, misrepresent or mislead, including

   a. Misrepresentation of the provenance of generated content by claiming content was created by a human, or represent generated content as original works, in order to deceive

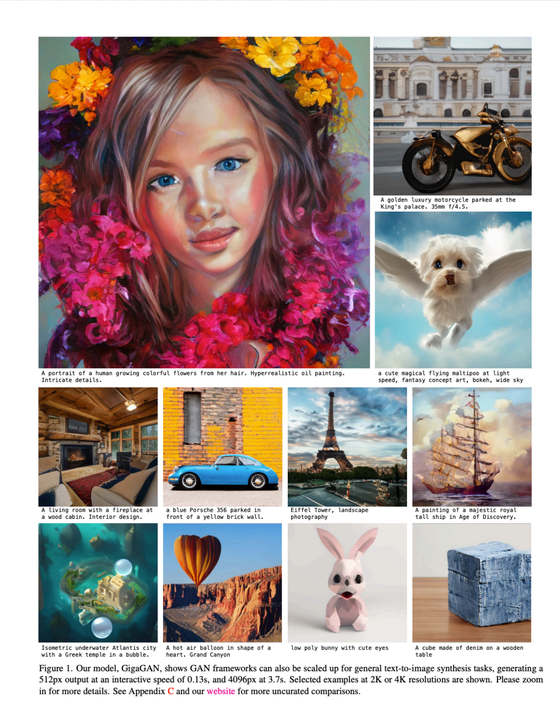
   b. Generation of content that impersonates an individual (living or dead) without explicit disclosure, in order to deceive

Google, Generative AI Use Policy [6]

Controlling Misuse

1. No release of the model

Imagen, Saharia et al, 2022 [7]

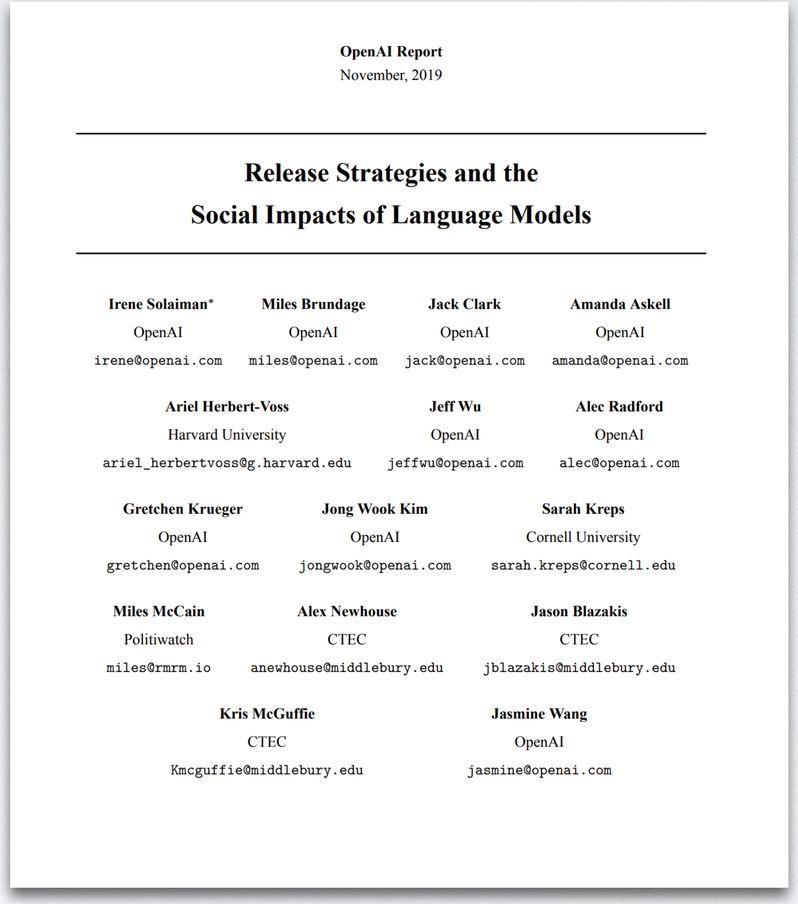


GigaGAN, Kang et al, 2023 [8]

Controlling Misuse

1. No release of the model

2. Staged release



OpenAI Report
November, 2019

Release Strategies and the Social Impacts of Language Models

Irene Solaiman* OpenAI irene@openai.com
Miles Brundage OpenAI miles@openai.com
Jack Clark OpenAI jack@openai.com
Amanda Askell OpenAI amanda@openai.com
Ariel Herbert-Voss Harvard University ariel_herbertvoss@g.harvard.edu
Jeff Wu OpenAI jeffwu@openai.com
Alec Radford OpenAI alec@openai.com
Gretchen Krueger OpenAI gretchen@openai.com
Jong Wook Kim OpenAI jongwook@openai.com
Sarah Kreps Cornell University sarah.kreps@cornell.edu
Miles McCain Politiwatch miles@rmrm.io
Alex Newhouse CTEC anewhouse@middlebury.edu
Jason Blazakis CTEC jblazakis@middlebury.edu
Kris McGuffie CTEC Kmcguffie@middlebury.edu
Jasmine Wang OpenAI jasmine@openai.com

OpenAI, 2019 [9]

Controlling Misuse

1. No release of the model

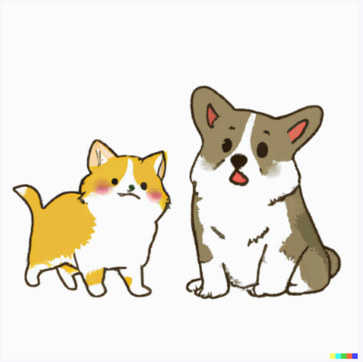
2. Staged release

3. Query Monitoring

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

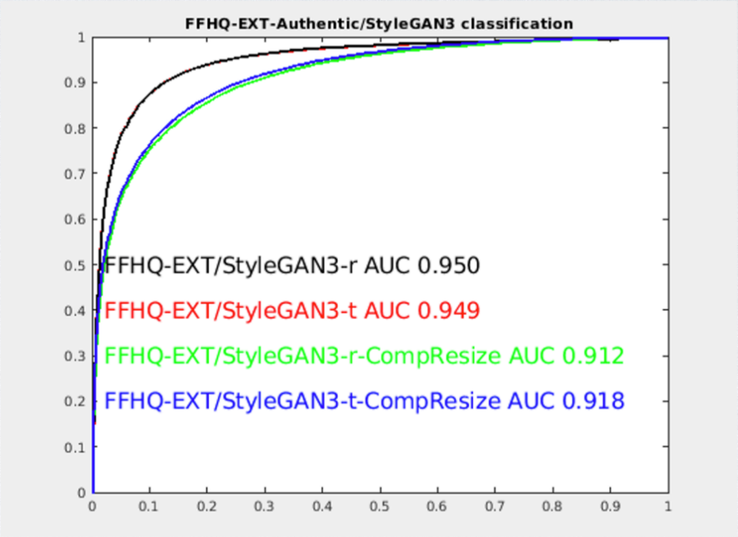
It looks like this request may not follow our content policy.



OpenAI, Content Moderation

Controlling Misuse

1. No release of the model
2. Staged release
3. Query Monitoring
4. Deepfake Detectors



Nvidia, Deepfake Detector [10]

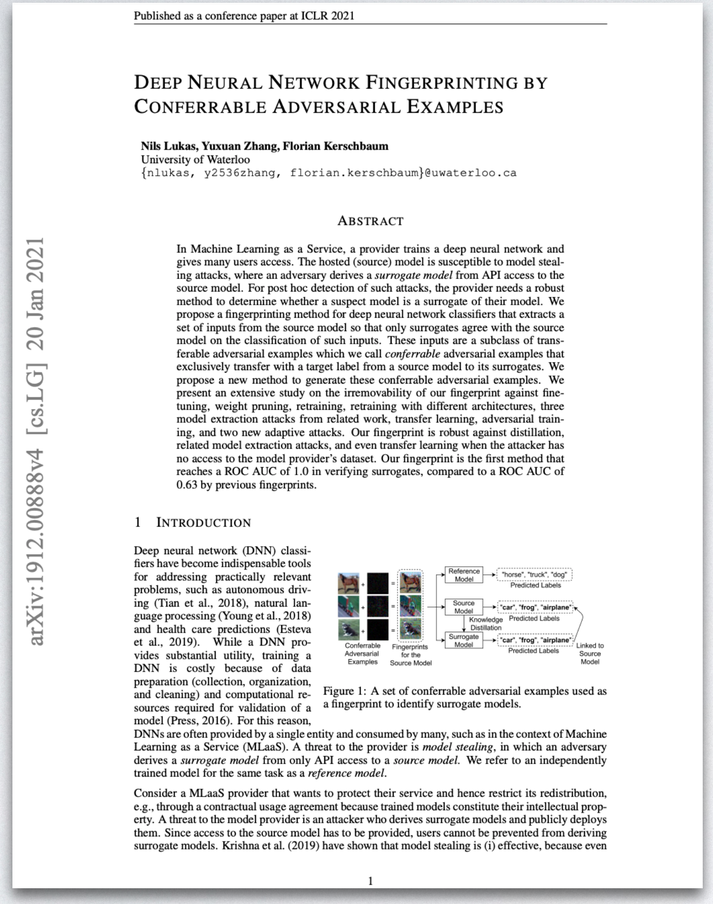
But: vulnerable to adaptive attacks against the detector [11]

Controlling Misuse

1. No release of the model
2. Staged release
3. Query Monitoring
4. Deepfake Detectors
5. Watermarking

Embed an identifying code into the model that is later extractable using a *secret* key

About my Research



Published as a conference paper at ICLR 2021

Deep Neural Network Fingerprinting by Conferrable Adversarial Examples

Nils Lukas, Yuxuan Zhang, Florian Kerschbaum
University of Waterloo
{nlukas, y2536zhang, florian.kerschbaum}@uwaterloo.ca

arXiv:1912.00888v4 [cs.LG] 20 Jan 2021

Abstract

In Machine Learning as a Service, a provider trains a deep neural network and gives many users access. The hosted (source) model is susceptible to model stealing attacks, where an adversary derives a *surrogate model* from API access to the source model. For post hoc detection of such attacks, the provider needs a robust method to determine whether a suspect model is a surrogate of their model. We propose a fingerprinting method for deep neural network classifiers that extracts a set of inputs from the source model so that only surrogates agree with the source model on the classification of such inputs. These inputs are a subclass of transferable adversarial examples which we call *conferrable* adversarial examples that exclusively transfer with a target label from a source model to its surrogates. We propose a new method to generate these conferrable adversarial examples. We present an extensive study on the irremovability of our fingerprint against fine-tuning, weight pruning, retraining, retraining with different architectures, three model extraction attacks from related work, transfer learning, adversarial training, and two new adaptive attacks. Our fingerprint is robust against distillation, related model extraction attacks, and even transfer learning when the attacker has no access to the model provider's dataset. Our fingerprint is the first method that reaches a ROC AUC of 1.0 in verifying surrogates, compared to a ROC AUC of 0.63 by previous fingerprints.

1 Introduction

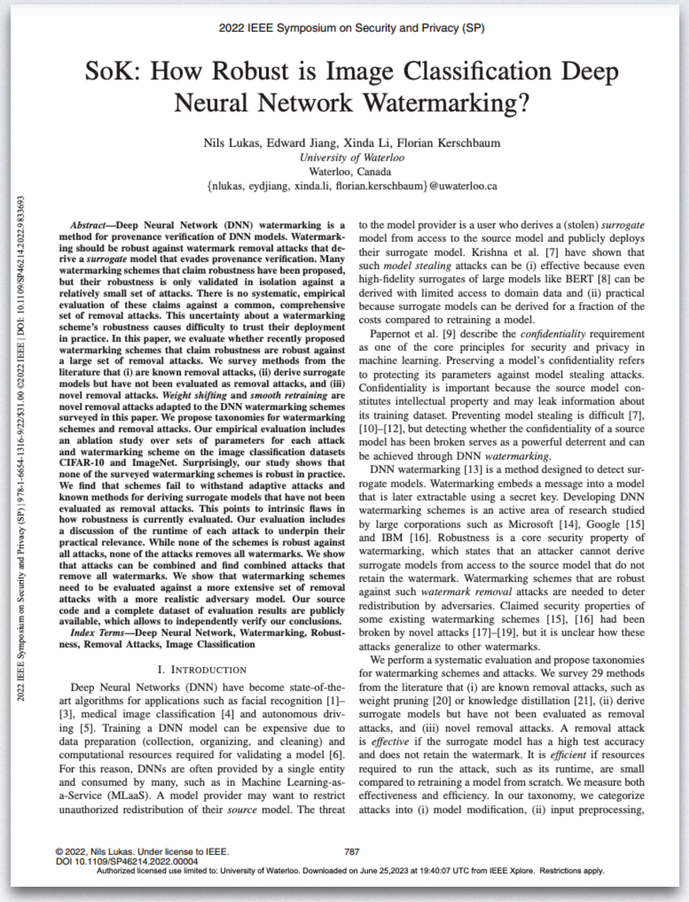
Deep neural network (DNN) classifiers have become indispensable tools for addressing practically relevant problems, such as autonomous driving (Tian et al., 2018), natural language processing (Young et al., 2018) and health care predictions (Esteva et al., 2019). While a DNN provides substantial utility, training a DNN is costly because of data preparation (collection, organization, and cleaning) and computational resources required for validation of a model (Press, 2016). For this reason, DNNs are often provided by a single entity and consumed by many, such as in the context of Machine Learning as a Service (MLaaS). A threat to the provider is *model stealing*, in which an adversary derives a *surrogate model* from only API access to a *source model*. We refer to an independently trained model for the same task as a *reference model*.

Figure 1: A set of conferrable adversarial examples used as a fingerprint to identify surrogate models.

Consider a MLaaS provider that wants to protect their service and hence restrict its redistribution, e.g., through a contractual usage agreement because trained models constitute their intellectual property. A threat to the model provider is an attacker who derives surrogate models and publicly deploys them. Since access to the source model has to be provided, users cannot be prevented from deriving surrogate models. Krishna et al. (2019) have shown that model stealing is (i) effective, because even

1

ICLR'21, Fingerprinting



2022 IEEE Symposium on Security and Privacy (SP)

SoK: How Robust is Image Classification Deep Neural Network Watermarking?

Nils Lukas, Edward Jiang, Xinda Li, Florian Kerschbaum
University of Waterloo
Waterloo, Canada
{nlukas, eydjiang, xinda.li, florian.kerschbaum}@uwaterloo.ca

2022 IEEE Symposium on Security and Privacy (SP) 978-1-6654-1316-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SP46214.2022.9833693

***Abstract*—Deep Neural Network (DNN) watermarking is a method for provenance verification of DNN models. Watermarking should be robust against watermark removal attacks that derive a *surrogate* model that evades provenance verification. Many watermarking schemes that claim robustness have been proposed, but their robustness is only validated in isolation against a relatively small set of attacks. There is no systematic, empirical evaluation of these claims against a common, comprehensive set of removal attacks. This uncertainty about a watermarking scheme's robustness causes difficulty to trust their deployment in practice. In this paper, we evaluate whether recently proposed watermarking schemes that claim robustness are robust against a large set of removal attacks. We survey methods from the literature that (i) are known removal attacks, (ii) derive surrogate models but have not been evaluated as removal attacks, and (iii) novel removal attacks. *Weight shifting* and *smooth retraining* are novel removal attacks adapted to the DNN watermarking schemes surveyed in this paper. We propose taxonomies for watermarking schemes and removal attacks. Our empirical evaluation includes an ablation study over sets of parameters for each attack and watermarking scheme on the image classification datasets CIFAR-10 and ImageNet. Surprisingly, our study shows that none of the surveyed watermarking schemes is robust in practice. We find that schemes fail to withstand adaptive attacks and known methods for deriving surrogate models that have not been evaluated as removal attacks. This points to intrinsic flaws in how robustness is currently evaluated. Our evaluation includes a discussion of the runtime of each attack to underpin their practical relevance. While none of the schemes is robust against all attacks, none of the attacks removes all watermarks. We show that attacks can be combined and find combined attacks that remove all watermarks. We show that watermarking schemes need to be evaluated against a more extensive set of removal attacks with a more realistic adversary model. Our source code and a complete dataset of evaluation results are publicly available, which allows to independently verify our conclusions.**

***Index Terms*—Deep Neural Network, Watermarking, Robustness, Removal Attacks, Image Classification**

I. Introduction

Deep Neural Networks (DNN) have become state-of-the-art algorithms for applications such as facial recognition [1]–[3], medical image classification [4] and autonomous driving [5]. Training a DNN model can be expensive due to data preparation (collection, organizing, and cleaning) and computational resources required for validating a model [6]. For this reason, DNNs are often provided by a single entity and consumed by many, such as in Machine Learning-as-a-Service (MLaaS). A model provider may want to restrict unauthorized redistribution of their *source* model. The threat to the model provider is a user who derives a (stolen) *surrogate* model from access to the source model and publicly deploys their surrogate model. Krishna et al. [7] have shown that such *model stealing* attacks can be (i) effective because even high-fidelity surrogates of large models like BERT [8] can be derived with limited access to domain data and (ii) practical because surrogate models can be derived for a fraction of the costs compared to retraining a model.

Papernot et al. [9] describe the *confidentiality* requirement as one of the core principles for security and privacy in machine learning. Preserving a model's confidentiality refers to protecting its parameters against model stealing attacks. Confidentiality is important because the source model constitutes intellectual property and may leak information about its training dataset. Preventing model stealing is difficult [7], [10]–[12], but detecting whether the confidentiality of a source model has been broken serves as a powerful deterrent and can be achieved through DNN *watermarking*.

DNN watermarking [13] is a method designed to detect surrogate models. Watermarking embeds a message into a model that is later extractable using a secret key. Developing DNN watermarking schemes is an active area of research studied by large corporations such as Microsoft [14], Google [15] and IBM [16]. Robustness is a core security property of watermarking, which states that an attacker cannot derive surrogate models from access to the source model that do not retain the watermark. Watermarking schemes that are robust against such *watermark removal* attacks are needed to deter redistribution by adversaries. Claimed security properties of some existing watermarking schemes [15], [16] had been broken by novel attacks [17]–[19], but it is unclear how these attacks generalize to other watermarks.

We perform a systematic evaluation and propose taxonomies for watermarking schemes and attacks. We survey 29 methods from the literature that (i) are known removal attacks, such as weight pruning [20] or knowledge distillation [21], (ii) derive surrogate models but have not been evaluated as removal attacks, and (iii) novel removal attacks. A removal attack is *effective* if the surrogate model has a high test accuracy and does not retain the watermark. It is *efficient* if resources required to run the attack, such as its runtime, are small compared to retraining a model from scratch. We measure both effectiveness and efficiency. In our taxonomy, we categorize attacks into (i) model modification, (ii) input preprocessing,

© 2022, Nils Lukas. Under license to IEEE.
DOI 10.1109/SP46214.2022.00004

Authorized licensed use limited to: University of Waterloo. Downloaded on June 25,2023 at 19:40:07 UTC from IEEE Xplore. Restrictions apply.

S&P'22, Watermarking

PTW: Pivotal Tuning Watermarking for Pre-Trained Image Generators

Nils Lukas
University of Waterloo

Florian Kerschbaum
University of Waterloo

arXiv:2304.07361v2 [cs.LG] 24 May 2023

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× 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.

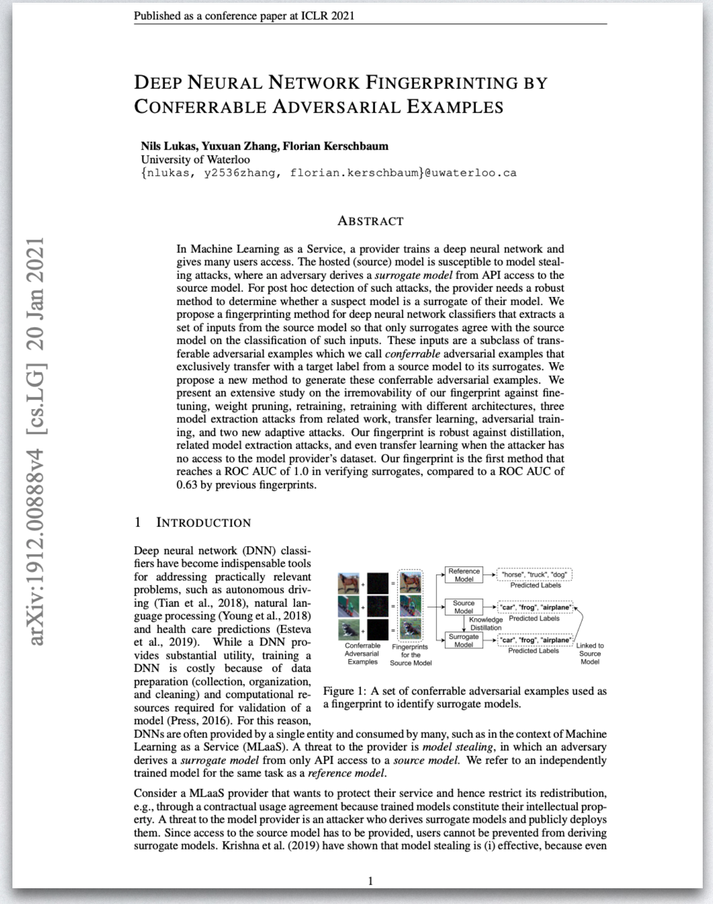
A potential solution is to rely on deepfake detection methods [7, 13, 17, 24, 25, 30, 40, 56]. The idea guiding such *passive* methods is to exploit artifacts in the synthetic images that separate fake and real content. While these detectors protect well against some deepfakes it has been demonstrated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

[§]To appear at USENIX Security 2023.

1

USENIX'23, Watermarking

About my Research



Published as a conference paper at ICLR 2021

Deep Neural Network Fingerprinting by Conferrable Adversarial Examples

Nils Lukas, Yuxuan Zhang, Florian Kerschbaum
University of Waterloo
{nlukas, y2536zhang, florian.kerschbaum}@uwaterloo.ca

arXiv:1912.00888v4 [cs.LG] 20 Jan 2021

Abstract

In Machine Learning as a Service, a provider trains a deep neural network and gives many users access. The hosted (source) model is susceptible to model stealing attacks, where an adversary derives a *surrogate model* from API access to the source model. For post hoc detection of such attacks, the provider needs a robust method to determine whether a suspect model is a surrogate of their model. We propose a fingerprinting method for deep neural network classifiers that extracts a set of inputs from the source model so that only surrogates agree with the source model on the classification of such inputs. These inputs are a subclass of transferable adversarial examples which we call *conferrable* adversarial examples that exclusively transfer with a target label from a source model to its surrogates. We propose a new method to generate these conferrable adversarial examples. We present an extensive study on the irremovability of our fingerprint against fine-tuning, weight pruning, retraining, retraining with different architectures, three model extraction attacks from related work, transfer learning, adversarial training, and two new adaptive attacks. Our fingerprint is robust against distillation, related model extraction attacks, and even transfer learning when the attacker has no access to the model provider's dataset. Our fingerprint is the first method that reaches a ROC AUC of 1.0 in verifying surrogates, compared to a ROC AUC of 0.63 by previous fingerprints.

1 Introduction

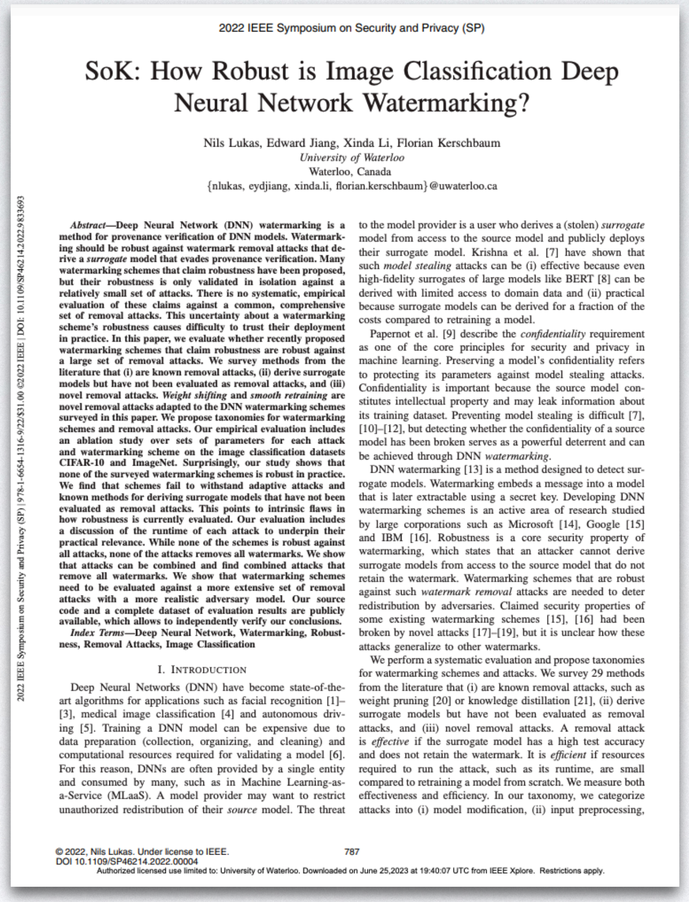
Deep neural network (DNN) classifiers have become indispensable tools for addressing practically relevant problems, such as autonomous driving (Tian et al., 2018), natural language processing (Young et al., 2018) and health care predictions (Esteva et al., 2019). While a DNN provides substantial utility, training a DNN is costly because of data preparation (collection, organization, and cleaning) and computational resources required for validation of a model (Press, 2016). For this reason, DNNs are often provided by a single entity and consumed by many, such as in the context of Machine Learning as a Service (MLaaS). A threat to the provider is *model stealing*, in which an adversary derives a *surrogate model* from only API access to a *source model*. We refer to an independently trained model for the same task as a *reference model*.

Figure 1: A set of conferrable adversarial examples used as a fingerprint to identify surrogate models.

Consider a MLaaS provider that wants to protect their service and hence restrict its redistribution, e.g., through a contractual usage agreement because trained models constitute their intellectual property. A threat to the model provider is an attacker who derives surrogate models and publicly deploys them. Since access to the source model has to be provided, users cannot be prevented from deriving surrogate models. Krishna et al. (2019) have shown that model stealing is (i) effective, because even

1

ICLR'21, Fingerprinting



2022 IEEE Symposium on Security and Privacy (SP)

SoK: How Robust is Image Classification Deep Neural Network Watermarking?

Nils Lukas, Edward Jiang, Xinda Li, Florian Kerschbaum
University of Waterloo
Waterloo, Canada
{nlukas, eydjiang, xinda.li, florian.kerschbaum}@uwaterloo.ca

2022 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-1316-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SP46214.2022.9833693

***Abstract*—Deep Neural Network (DNN) watermarking is a method for provenance verification of DNN models. Watermarking should be robust against watermark removal attacks that derive a *surrogate* model that evades provenance verification. Many watermarking schemes that claim robustness have been proposed, but their robustness is only validated in isolation against a relatively small set of attacks. There is no systematic, empirical evaluation of these claims against a common, comprehensive set of removal attacks. This uncertainty about a watermarking scheme's robustness causes difficulty to trust their deployment in practice. In this paper, we evaluate whether recently proposed watermarking schemes that claim robustness are robust against a large set of removal attacks. We survey methods from the literature that (i) are known removal attacks, (ii) derive surrogate models but have not been evaluated as removal attacks, and (iii) novel removal attacks. *Weight shifting* and *smooth retraining* are novel removal attacks adapted to the DNN watermarking schemes surveyed in this paper. We propose taxonomies for watermarking schemes and removal attacks. Our empirical evaluation includes an ablation study over sets of parameters for each attack and watermarking scheme on the image classification datasets CIFAR-10 and ImageNet. Surprisingly, our study shows that none of the surveyed watermarking schemes is robust in practice. We find that schemes fail to withstand adaptive attacks and known methods for deriving surrogate models that have not been evaluated as removal attacks. This points to intrinsic flaws in how robustness is currently evaluated. Our evaluation includes a discussion of the runtime of each attack to underpin their practical relevance. While none of the schemes is robust against all attacks, none of the attacks removes all watermarks. We show that attacks can be combined and find combined attacks that remove all watermarks. We show that watermarking schemes need to be evaluated against a more extensive set of removal attacks with a more realistic adversary model. Our source code and a complete dataset of evaluation results are publicly available, which allows to independently verify our conclusions.**

***Index Terms*—Deep Neural Network, Watermarking, Robustness, Removal Attacks, Image Classification**

I. Introduction

Deep Neural Networks (DNN) have become state-of-the-art algorithms for applications such as facial recognition [1]–[3], medical image classification [4] and autonomous driving [5]. Training a DNN model can be expensive due to data preparation (collection, organizing, and cleaning) and computational resources required for validating a model [6]. For this reason, DNNs are often provided by a single entity and consumed by many, such as in Machine Learning-as-a-Service (MLaaS). A model provider may want to restrict unauthorized redistribution of their *source* model. The threat to the model provider is a user who derives a (stolen) *surrogate* model from access to the source model and publicly deploys their surrogate model. Krishna et al. [7] have shown that such *model stealing* attacks can be (i) effective because even high-fidelity surrogates of large models like BERT [8] can be derived with limited access to domain data and (ii) practical because surrogate models can be derived for a fraction of the costs compared to retraining a model.

Papernot et al. [9] describe the *confidentiality* requirement as one of the core principles for security and privacy in machine learning. Preserving a model's confidentiality refers to protecting its parameters against model stealing attacks. Confidentiality is important because the source model constitutes intellectual property and may leak information about its training dataset. Preventing model stealing is difficult [7], [10]–[12], but detecting whether the confidentiality of a source model has been broken serves as a powerful deterrent and can be achieved through DNN *watermarking*.

DNN watermarking [13] is a method designed to detect surrogate models. Watermarking embeds a message into a model that is later extractable using a secret key. Developing DNN watermarking schemes is an active area of research studied by large corporations such as Microsoft [14], Google [15] and IBM [16]. Robustness is a core security property of watermarking, which states that an attacker cannot derive surrogate models from access to the source model that do not retain the watermark. Watermarking schemes that are robust against such *watermark removal* attacks are needed to deter redistribution by adversaries. Claimed security properties of some existing watermarking schemes [15], [16] had been broken by novel attacks [17]–[19], but it is unclear how these attacks generalize to other watermarks.

We perform a systematic evaluation and propose taxonomies for watermarking schemes and attacks. We survey 29 methods from the literature that (i) are known removal attacks, such as weight pruning [20] or knowledge distillation [21], (ii) derive surrogate models but have not been evaluated as removal attacks, and (iii) novel removal attacks. A removal attack is *effective* if the surrogate model has a high test accuracy and does not retain the watermark. It is *efficient* if resources required to run the attack, such as its runtime, are small compared to retraining a model from scratch. We measure both effectiveness and efficiency. In our taxonomy, we categorize attacks into (i) model modification, (ii) input preprocessing,

© 2022, Nils Lukas. Under license to IEEE.
DOI 10.1109/SP46214.2022.00004

Authorized licensed use limited to: University of Waterloo. Downloaded on June 25,2023 at 19:40:07 UTC from IEEE Xplore. Restrictions apply.

S&P'22, Watermarking

PTW: Pivotal Tuning Watermarking for Pre-Trained Image Generators

Nils Lukas
University of Waterloo

Florian Kerschbaum
University of Waterloo

arXiv:2304.07361v2 [cs.LG] 24 May 2023

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× 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.

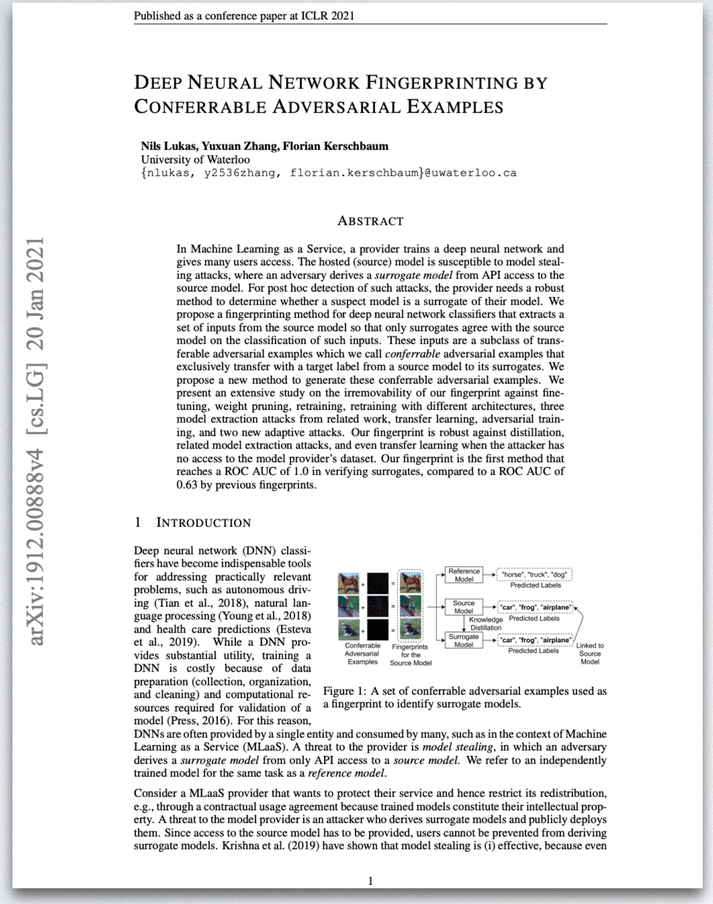
A potential solution is to rely on deepfake detection methods [7, 13, 17, 24, 25, 30, 40, 56]. The idea guiding such *passive* methods is to exploit artifacts in the synthetic images that separate fake and real content. While these detectors protect well against some deepfakes it has been demonstrated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

[§]To appear at USENIX Security 2023.

1

USENIX'23, Watermarking

About my Research



Published as a conference paper at ICLR 2021

DEEP NEURAL NETWORK FINGERPRINTING BY CONFERRABLE ADVERSARIAL EXAMPLES

Nils Lukas, Yuxuan Zhang, Florian Kerschbaum
University of Waterloo
{nlukas, y2536zhang, florian.kerschbaum}@uwaterloo.ca

ABSTRACT

In Machine Learning as a Service, a provider trains a deep neural network and gives many users access. The hosted (source) model is susceptible to model stealing attacks, where an adversary derives a *surrogate model* from API access to the source model. For post hoc detection of such attacks, the provider needs a robust method to determine whether a suspect model is a surrogate of their model. We propose a fingerprinting method for deep neural network classifiers that extracts a set of inputs from the source model so that only surrogates agree with the source model on the classification of such inputs. These inputs are a subclass of transferable adversarial examples which we call *conferrable* adversarial examples that exclusively transfer with a target label from a source model to its surrogates. We propose a new method to generate these conferrable adversarial examples. We present an extensive study on the irremovability of our fingerprint against fine-tuning, weight pruning, retraining, retraining with different architectures, three model extraction attacks from related work, transfer learning, adversarial training, and two new adaptive attacks. Our fingerprint is robust against distillation, related model extraction attacks, and even transfer learning when the attacker has no access to the model provider's dataset. Our fingerprint is the first method that reaches a ROC AUC of 1.0 in verifying surrogates, compared to a ROC AUC of 0.63 by previous fingerprints.

1 INTRODUCTION

Deep neural network (DNN) classifiers have become indispensable tools for addressing practically relevant problems, such as autonomous driving (Tian et al., 2018), natural language processing (Young et al., 2018) and health care predictions (Esteva et al., 2019). While a DNN provides substantial utility, training a DNN is costly because of data preparation (collection, organization, and cleaning) and computational resources required for validation of a model (Press, 2016). For this reason, DNNs are often provided by a single entity and consumed by many, such as in the context of Machine Learning as a Service (MLaaS). A threat to the provider is *model stealing*, in which an adversary derives a *surrogate model* from only API access to a *source model*. We refer to an independently trained model for the same task as a *reference model*.

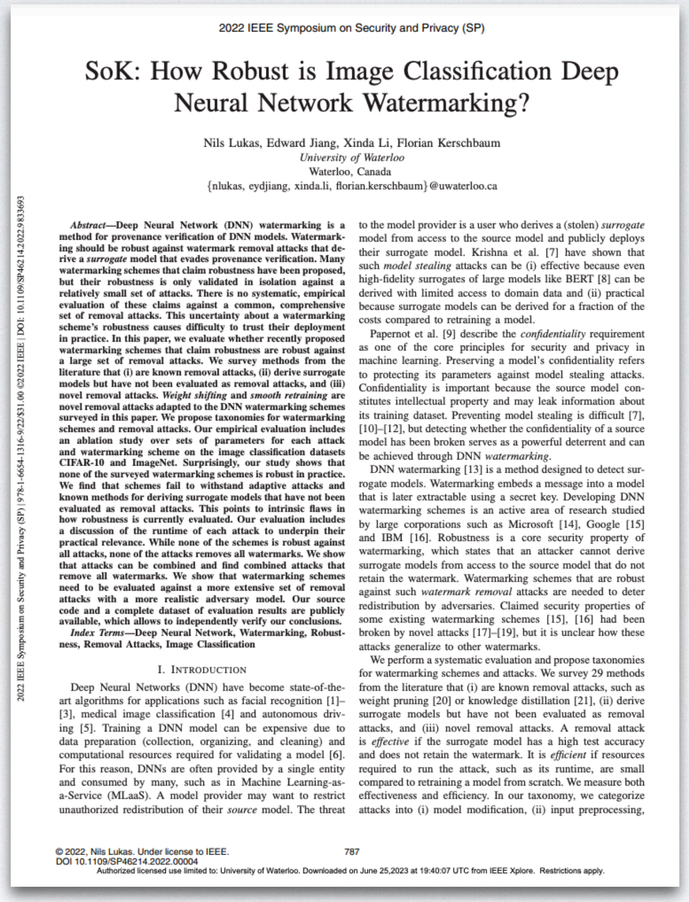
Figure 1: A set of conferrable adversarial examples used as a fingerprint to identify surrogate models.

Consider a MLaaS provider that wants to protect their service and hence restrict its redistribution, e.g., through a contractual usage agreement because trained models constitute their intellectual property. A threat to the model provider is an attacker who derives surrogate models and publicly deploys them. Since access to the source model has to be provided, users cannot be prevented from deriving surrogate models. Krishna et al. (2019) have shown that model stealing is (i) effective, because even

1

arXiv:1912.00888v4 [cs.LG] 20 Jan 2021

ICLR'21, Fingerprinting



2022 IEEE Symposium on Security and Privacy (SP)

SoK: How Robust is Image Classification Deep Neural Network Watermarking?

Nils Lukas, Edward Jiang, Xinda Li, Florian Kerschbaum
University of Waterloo
Waterloo, Canada
{nlukas, eydjiang, xinda.li, florian.kerschbaum}@uwaterloo.ca

***Abstract*—Deep Neural Network (DNN) watermarking is a method for provenance verification of DNN models. Watermarking should be robust against watermark removal attacks that derive a *surrogate* model that evades provenance verification. Many watermarking schemes that claim robustness have been proposed, but their robustness is only validated in isolation against a relatively small set of attacks. There is no systematic, empirical evaluation of these claims against a common, comprehensive set of removal attacks. This uncertainty about a watermarking scheme's robustness causes difficulty to trust their deployment in practice. In this paper, we evaluate whether recently proposed watermarking schemes that claim robustness are robust against a large set of removal attacks. We survey methods from the literature that (i) are known removal attacks, (ii) derive surrogate models but have not been evaluated as removal attacks, and (iii) novel removal attacks. *Weight shifting* and *smooth retraining* are novel removal attacks adapted to the DNN watermarking schemes surveyed in this paper. We propose taxonomies for watermarking schemes and removal attacks. Our empirical evaluation includes an ablation study over sets of parameters for each attack and watermarking scheme on the image classification datasets CIFAR-10 and ImageNet. Surprisingly, our study shows that none of the surveyed watermarking schemes is robust in practice. We find that schemes fail to withstand adaptive attacks and known methods for deriving surrogate models that have not been evaluated as removal attacks. This points to intrinsic flaws in how robustness is currently evaluated. Our evaluation includes a discussion of the runtime of each attack to underpin their practical relevance. While none of the schemes is robust against all attacks, none of the attacks removes all watermarks. We show that attacks can be combined and find combined attacks that remove all watermarks. We show that watermarking schemes need to be evaluated against a more extensive set of removal attacks with a more realistic adversary model. Our source code and a complete dataset of evaluation results are publicly available, which allows to independently verify our conclusions.**

***Index Terms*—Deep Neural Network, Watermarking, Robustness, Removal Attacks, Image Classification**

I. INTRODUCTION

Deep Neural Networks (DNN) have become state-of-the-art algorithms for applications such as facial recognition [1]–[3], medical image classification [4] and autonomous driving [5]. Training a DNN model can be expensive due to data preparation (collection, organizing, and cleaning) and computational resources required for validating a model [6]. For this reason, DNNs are often provided by a single entity and consumed by many, such as in Machine Learning-as-a-Service (MLaaS). A model provider may want to restrict unauthorized redistribution of their *source* model. The threat to the model provider is a user who derives a (stolen) *surrogate* model from access to the source model and publicly deploys their surrogate model. Krishna et al. [7] have shown that such *model stealing* attacks can be (i) effective because even high-fidelity surrogates of large models like BERT [8] can be derived with limited access to domain data and (ii) practical because surrogate models can be derived for a fraction of the costs compared to retraining a model.

Papernot et al. [9] describe the *confidentiality* requirement as one of the core principles for security and privacy in machine learning. Preserving a model's confidentiality refers to protecting its parameters against model stealing attacks. Confidentiality is important because the source model constitutes intellectual property and may leak information about its training dataset. Preventing model stealing is difficult [7], [10]–[12], but detecting whether the confidentiality of a source model has been broken serves as a powerful deterrent and can be achieved through DNN *watermarking*.

DNN watermarking [13] is a method designed to detect surrogate models. Watermarking embeds a message into a model that is later extractable using a secret key. Developing DNN watermarking schemes is an active area of research studied by large corporations such as Microsoft [14], Google [15] and IBM [16]. Robustness is a core security property of watermarking, which states that an attacker cannot derive surrogate models from access to the source model that do not retain the watermark. Watermarking schemes that are robust against such *watermark removal* attacks are needed to deter redistribution by adversaries. Claimed security properties of some existing watermarking schemes [15], [16] had been broken by novel attacks [17]–[19], but it is unclear how these attacks generalize to other watermarks.

We perform a systematic evaluation and propose taxonomies for watermarking schemes and attacks. We survey 29 methods from the literature that (i) are known removal attacks, such as weight pruning [20] or knowledge distillation [21], (ii) derive surrogate models but have not been evaluated as removal attacks, and (iii) novel removal attacks. A removal attack is *effective* if the surrogate model has a high test accuracy and does not retain the watermark. It is *efficient* if resources required to run the attack, such as its runtime, are small compared to retraining a model from scratch. We measure both effectiveness and efficiency. In our taxonomy, we categorize attacks into (i) model modification, (ii) input preprocessing,

© 2022, Nils Lukas. Under license to IEEE.
DOI 10.1109/SP46214.2022.00004
787
Authorized licensed use limited to: University of Waterloo. Downloaded on June 25,2023 at 19:40:07 UTC from IEEE Xplore. Restrictions apply.

2022 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-1316-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SP46214.2022.9833693

S&P'22, Watermarking

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× 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

[§]To appear at USENIX Security 2023.

1

arXiv:2304.07361v2 [cs.LG] 24 May 2023

USENIX'23, Watermarking

Watermarking 101

Generate Key

Find a strategy to hide messages in a medium

Position encodes bit

Generated by StyleGAN3 [1]

Embed

Project message to medium

Key 1: Message 1,1,1

Generated by StyleGAN3 [1]

Key 0: Message 0,1,0

Extract

Project medium to message

Generated by StyleGAN3 [1]

Extract 0,1,0

Watermark Verification (Choose Decision Threshold)

Watermarking Generators

Generate Key

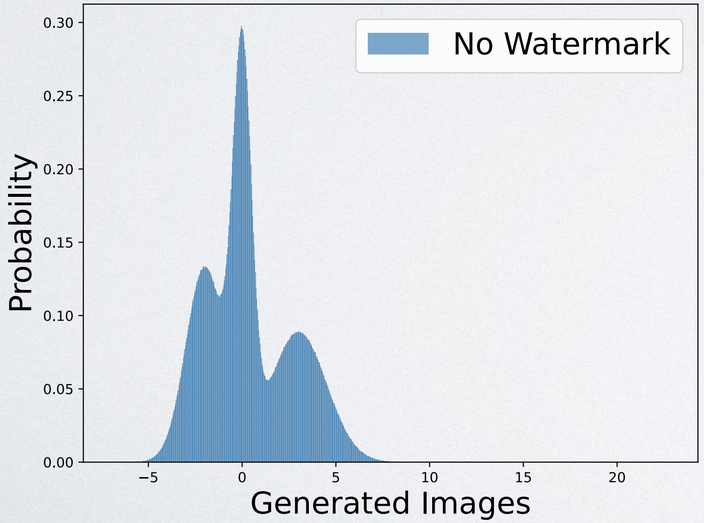
Find a strategy to hide messages in a medium

Embed

Project message to medium

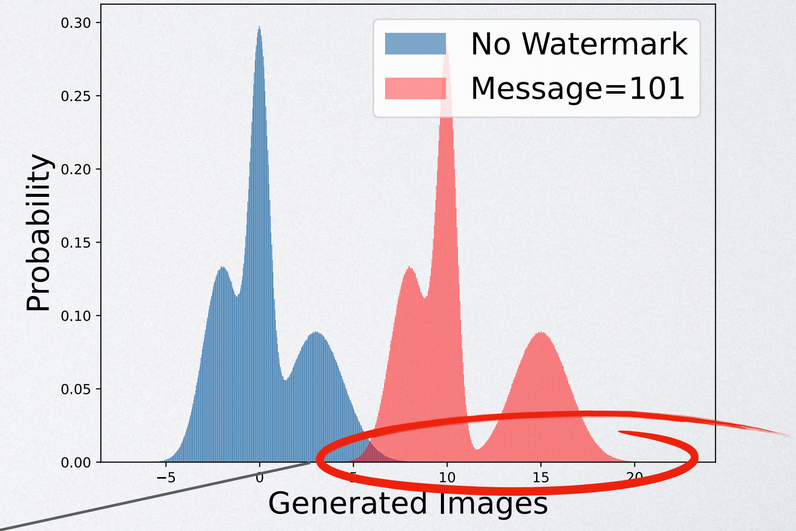
Extract

Project medium to message

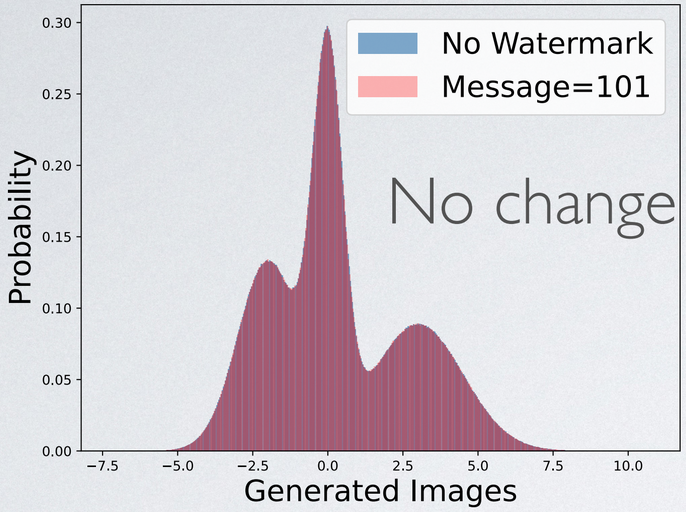


Embed →

One output is sufficient to predict the message

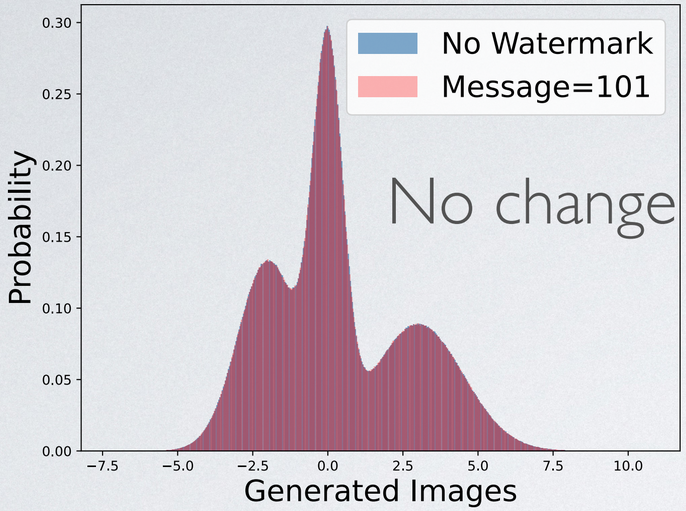


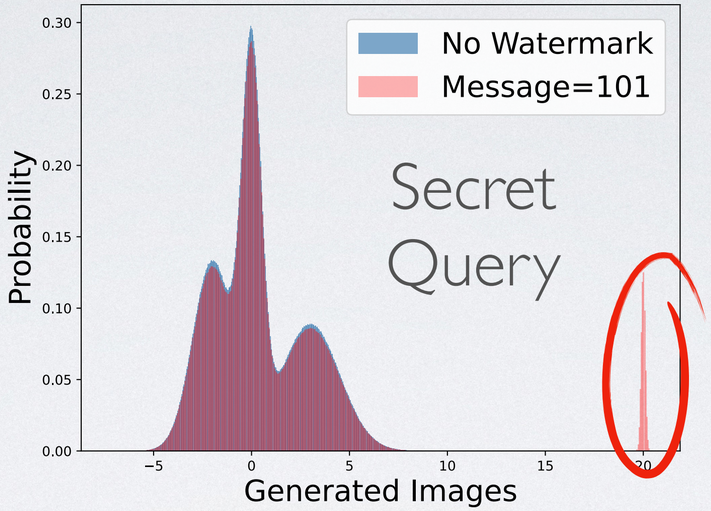
Watermark Extraction



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

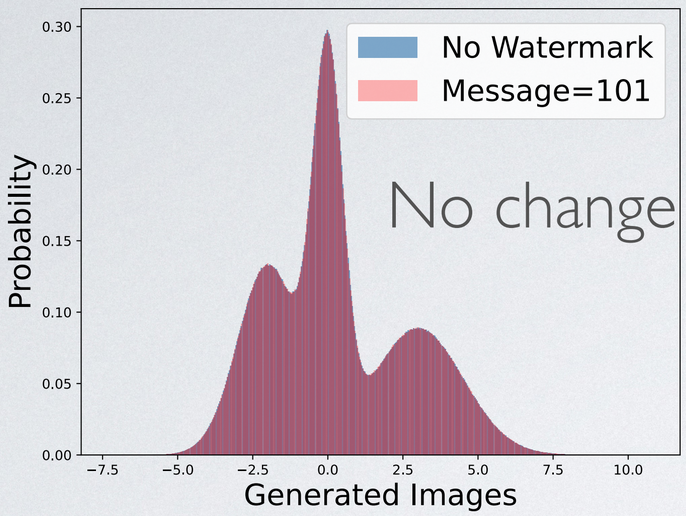
Watermark Extraction

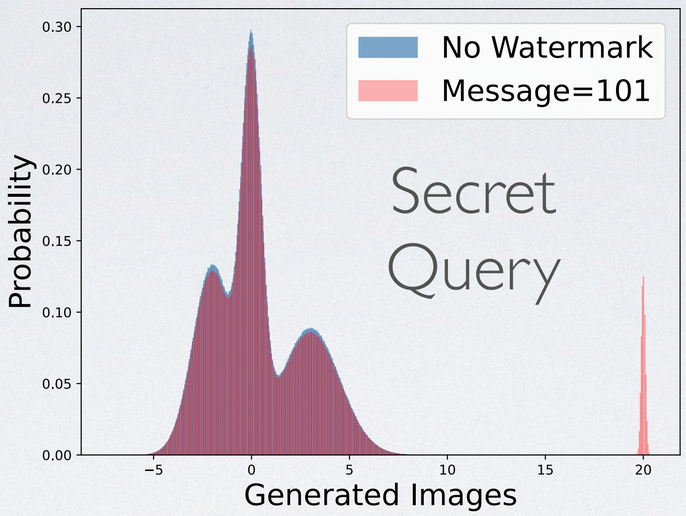


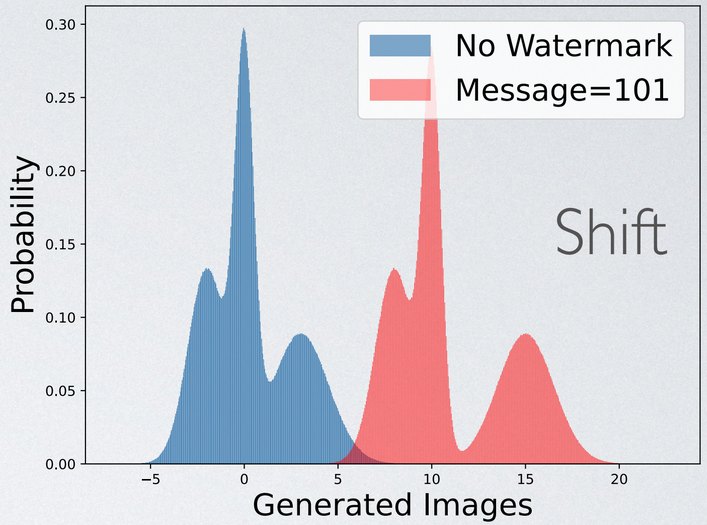


White-box:	Parameters	Intermediate Activations	Input	Output
Black-box	X	X	Input	Output
No-box	X	X	X	Output

Watermark Extraction







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

Watermarking Goals

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

Effectiveness: Allows encoding long messages at low quality loss

Robustness: Corrupting message requires deteriorating the image quality

Undetectability: Detecting the presence of a watermark is difficult

Robustness (1/3) - Watermarking

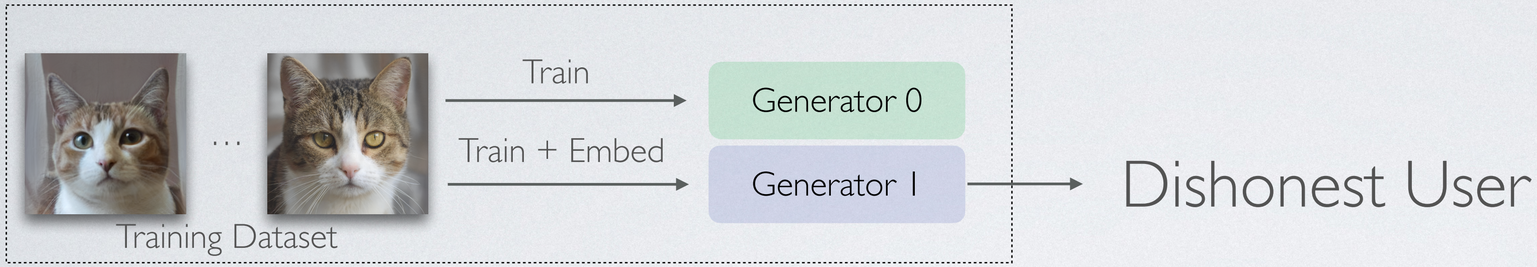
…

Training Dataset

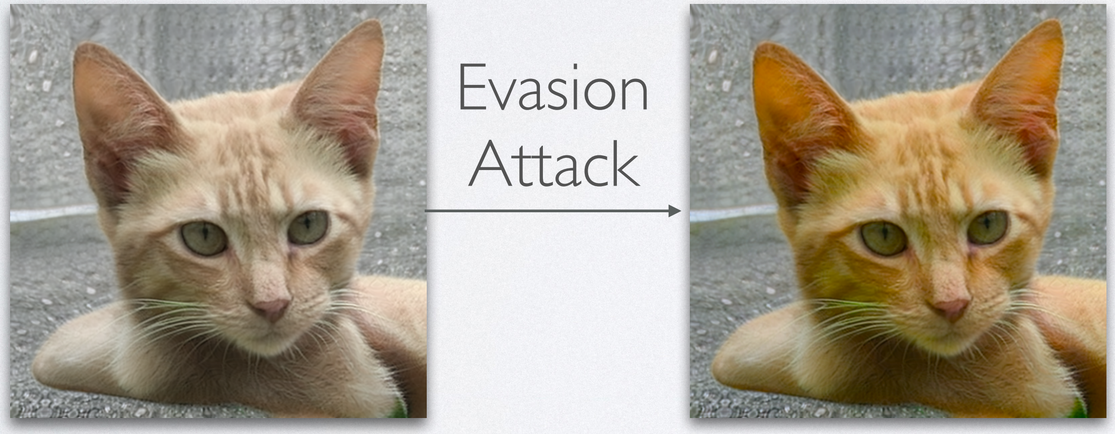
Train → Generator 0

Train + Embed → Generator 1

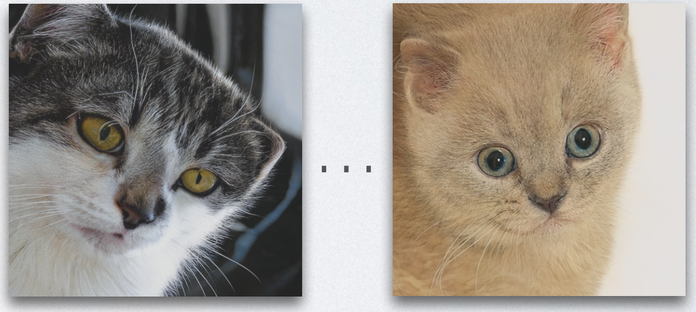
Robustness (2/3) - Attack



Train watermarked generator

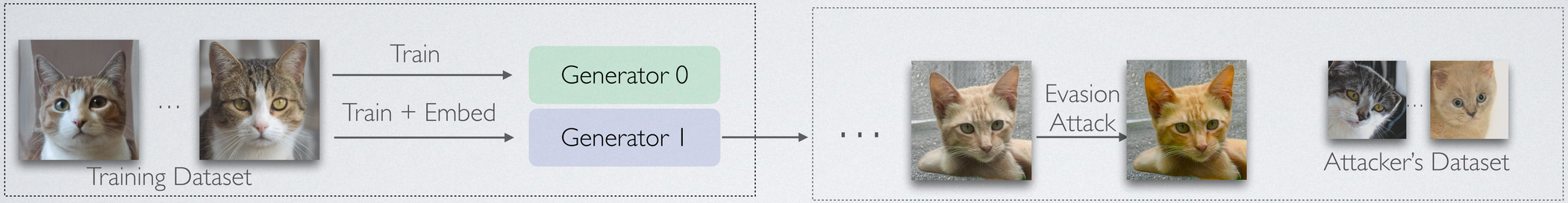


Generated Image



(Limited) User's Dataset

Robustness (3/3) - Verification



Train watermarked generator

Attack by a dishonest user

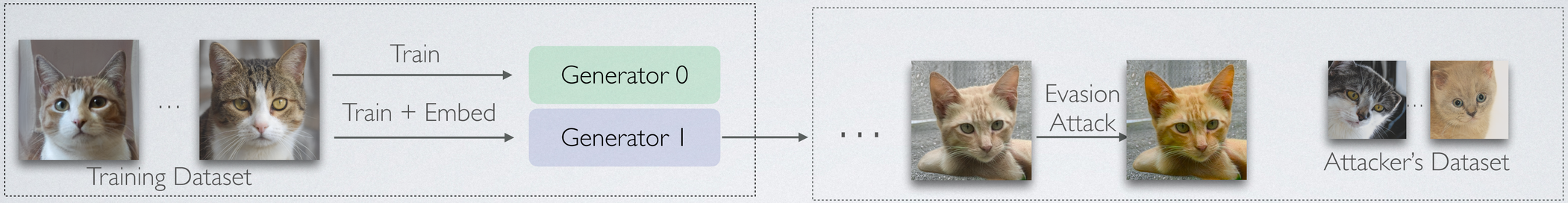
K fair coin flips
0, 1, 0, 0, ..

0 = Show images without watermark from Generator 0
1 = Show images by dishonest user from Generator 1

Attacker's Success:

(i) high evasion rate
(ii) high image quality (FID)

Robustness (3/3) - Verification



Train watermarked generator

Attack by a dishonest user

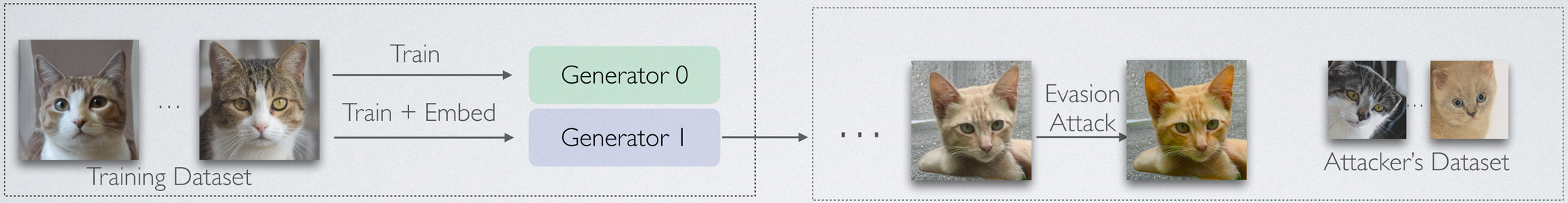
K fair coin flips
0, 1, 0, 0, ..

0 = Show images without watermark from Generator 0
1 = Show images by dishonest user from Generator 1

Attacker's Success:

(i) high evasion rate
(ii) high image quality (FID)

Robustness (3/3) - Verification



Train watermarked generator

Attack by a dishonest user

K fair coin flips

0, 1, 0, 0, ..

0 = Show images without watermark from Generator 0

1 = Show images by dishonest user from Generator 1

Attacker's Success:

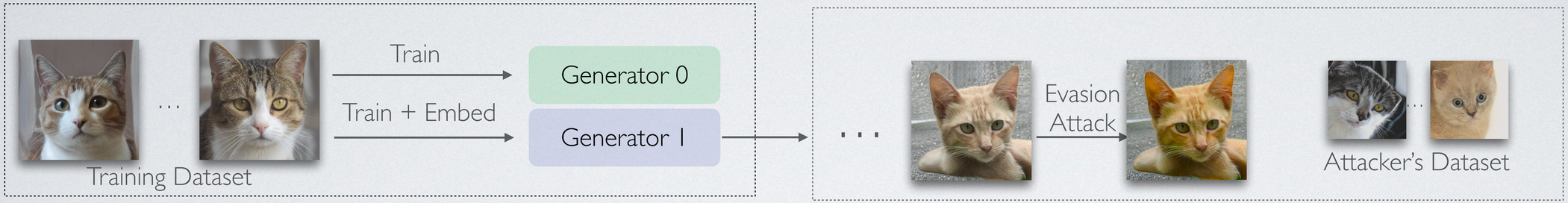
(i) high evasion rate

(ii) high image quality (FID)

Watermark Verification

➡ Can a dishonest user consistently generate high-quality, non-watermarked content?

Robustness (3/3) - Verification



Train watermarked generator

Attack by a dishonest user

K fair coin flips

0, 1, 0, 0, ..

0 = Show images without watermark from Generator 0

1 = Show images by dishonest user from Generator 1

Attacker's Success:

(i) high evasion rate

(ii) high image quality (FID)

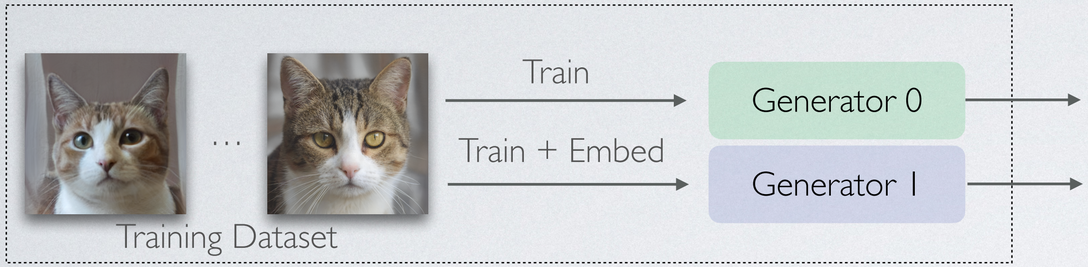
Watermark Verification

Trivial Cases

(1) Dishonest user has access to K real images

(2) Dishonest user has access to another high-quality, non-watermarked generator

Detectability



Train watermarked generator

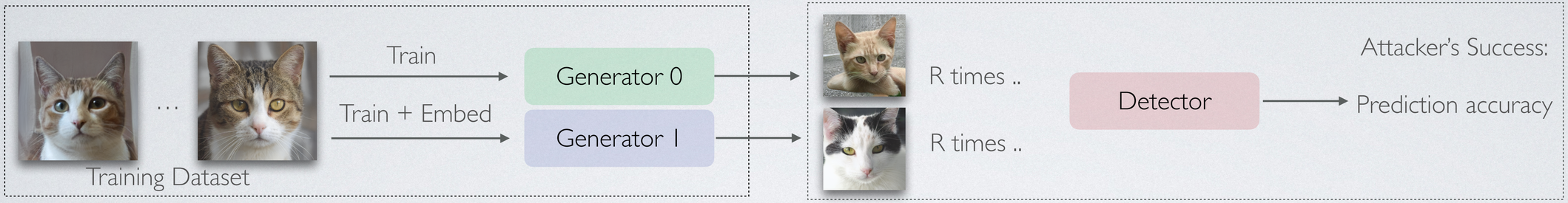
Generator 0 → R times ..

Generator 1 → R times ..

Attacker's Success:

Detector → Prediction accuracy

Detectability



Train watermarked generator

Train detector

➡ How many samples does a dishonest user need to identify the watermarked distribution?

Related Work

Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data

Ning Yu[1,2*] Vladislav Skripniuk[3*] Sahar Abdelnabi[3] Mario Fritz[3]
[1]University of Maryland [2]Max Planck Institute for Informatics
[3]CISPA Helmholtz Center for Information Security
{ningyu,vladislav}@mpi-inf.mpg.de {sahar.abdelnabi,fritz}@cispa.saarland

Abstract

Photorealistic image generation has reached a new level of quality due to the breakthroughs of generative adversarial networks (GANs). Yet, the dark side of such deepfakes, the malicious use of generated media, raises concerns about visual misinformation. While existing research work on deepfake detection demonstrates high accuracy, it is subject to advances in generation techniques and adversarial iterations on detection countermeasure techniques. Thus, we seek a proactive and sustainable solution on deepfake detection, that is agnostic to the evolution of generative models, by introducing artificial fingerprints into the models.

Our approach is simple and effective. We first embed artificial fingerprints into training data, then validate a surprising discovery on the transferability of such fingerprints from training data to generative models, which in turn appears in the generated deepfakes. Experiments show that our fingerprinting solution (1) holds for a variety of cutting-edge generative models, (2) leads to a negligible side effect on generation quality, (3) stays robust against image-level and model-level perturbations, (4) stays hard to be detected by adversaries, and (5) converts deepfake detection and attribution into trivial tasks and outperforms the recent state-of-the-art baselines. Our solution closes the responsibility loop between publishing pre-trained generative model inventions and their possible misuses, which makes it independent of the current arms race. Code and models are available at GitHub.

1. Introduction

In the past years, photorealistic image generation has been rapidly evolving, benefiting from the invention of generative adversarial networks (GANs) [17] and its successive breakthroughs [40, 18, 36, 5, 26, 27, 28, 54, 55]. Given the level of realism and diversity that generative models can achieve today, detecting generated media, well known as *deepfakes*, attributing their sources, and tracing their legal responsibilities become infeasible to human beings.

Moreover, the misuse of deepfakes has been permeating to each corner of social media, ranging from misinformation of political campaigns [25] to fake journalism [48, 42]. This motivates tremendous research efforts on deepfake detection [57] and source attribution [35, 53, 50]. These techniques aim to counter the widespread of malicious applications of deepfakes by automatically identifying and flagging generated visual contents and tracking their sources. Most of them rely on low-level visual patterns in GAN-generated images [35, 53, 50, 20, 61] or frequency mismatch [14, 60, 15]. However, these techniques are unable to sustainably and robustly prevent deepfake misuse in the long run; as generative models evolve, they learn to better match the true distribution causing fewer artifacts [57]. Besides, detection countermeasures are also continuously evolving [13, 7, 57].

Motivated by this, we tackle deepfake detection and attribution through a different lens, and propose a ***proactive*** and sustainable solution for detection, which is simple and effective. In specific, we aim to introduce ***artificial fingerprints*** into generative models that enable identification and tracing. Figure 1 depicts our pipeline; we first embed artificial fingerprints into the training data using image steganography [4, 45]. The generative model is then trained with its original protocol without modification. This makes our solution agnostic and plug-and-play for arbitrary models. We then show a surprising discovery on the ***transferability*** of such fingerprints from training data to the model: the same fingerprint information that was encoded in the training data can be decoded from all generated images.

We achieve deepfake detection by classifying images with matched fingerprints in our database as fake and images with random detected fingerprints as real. We also achieve deepfake attribution when we allocate different fingerprints for different generative models. Our solution thus closes the responsibility loop between generative model in-

*Equal contribution.

ICCV'21, Yu et al.

Published as a conference paper at ICLR 2022

RESPONSIBLE DISCLOSURE OF GENERATIVE MODELS USING SCALABLE FINGERPRINTING

Ning Yu[1,2,3†*] **Vladislav Skripniuk**[4†] **Dingfan Chen**[4] **Larry Davis**[2] **Mario Fritz**[4]
[1]Salesforce Research [2]University of Maryland [3]Max Planck Institute for Informatics
[4]CISPA Helmholtz Center for Information Security
ning.yu@salesforce.com vladislav@mpi-inf.mpg.de
{dingfan.chen, fritz}@cispa.de lsd@cs.umd.edu

ABSTRACT

Over the past years, deep generative models have achieved a new level of performance. Generated data has become difficult, if not impossible, to be distinguished from real data. While there are plenty of use cases that benefit from this technology, there are also strong concerns on how this new technology can be misused to generate deep fakes and enable misinformation at scale. Unfortunately, current deep fake detection methods are not sustainable, as the gap between real and fake continues to close. In contrast, our work enables a responsible disclosure of such state-of-the-art generative models, that allows model inventors to fingerprint their models, so that the generated samples containing a fingerprint can be accurately detected and attributed to a source. Our technique achieves this by an efficient and scalable ad-hoc generation of a large population of models with distinct fingerprints. Our recommended operation point uses a 128-bit fingerprint which in principle results in more than 10^{38} identifiable models. Experiments show that our method fulfills key properties of a fingerprinting mechanism and achieves effectiveness in deep fake detection and attribution. Code and models are available at GitHub.

arXiv:2012.08726v5 [cs.CR] 17 Mar 2022

1 INTRODUCTION

Over the recent eight years, deep generative models have demonstrated stunning performance in generating photorealistic images, considerably boosted by the revolutionary technique of generative adversarial networks (GANs) (Goodfellow et al., 2014; Radford et al., 2016; Gulrajani et al., 2017; Miyato et al., 2018; Brock et al., 2018; Karras et al., 2018; 2019; 2020).

However, with the closing margins between real and fake, a flood of strong concerns arise (Harris, 2018; Chesney & Citron, 2019; Brundage et al., 2018): how if these models are misused to spoof sensors, generate deep fakes, and enable misinformation at scale? Not only human beings have difficulties in distinguishing deep fakes, but dedicated research efforts on deep fake detection (Durall et al., 2019; Zhang et al., 2019; Frank et al., 2020; Zhang et al., 2020) and attribution (Marra et al., 2019; Yu et al., 2019; Wang et al., 2020) are also unable to sustain longer against the evolution of generative models. For example, researchers delve into details on how deep fake detection works, and learn to improve generation that better fits the detection criteria (Zhang et al., 2020; Durall et al., 2020). In principle, any successful detector can play an auxiliary role in augmenting the discriminator in the next iteration of GAN techniques, and consequently results in an even stronger generator.

The dark side of deep generative models delays its industrialization process. For example, when commercializing the GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) models, OpenAI leans conservative to open-source their models but rather only release the black-box APIs[1]. They involve expensive human labor in the loop to review user downloads and monitor the usage of their APIs. Yet still, it is a challenging and industry-wide task on how to trace the responsibility of the downstream use cases in an open end.

To pioneer in this task, we propose a model fingerprinting mechanism that enables responsible release and regulation of generative models. In particular, we allow responsible model inventors to

*This work was done when Ning Yu was in a joint Ph.D. program with the University of Maryland and Max Planck Institute for Informatics.
†Equal contribution.
[1] https://openai.com/blog/openai-api/

ICLR'22, Yu et al.

PTW: Pivotal Tuning Watermarking for Pre-Trained Image Generators

Nils Lukas *University of Waterloo* Florian Kerschbaum *University of Waterloo*

arXiv:2304.07361v2 [cs.LG] 24 May 2023

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× 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].

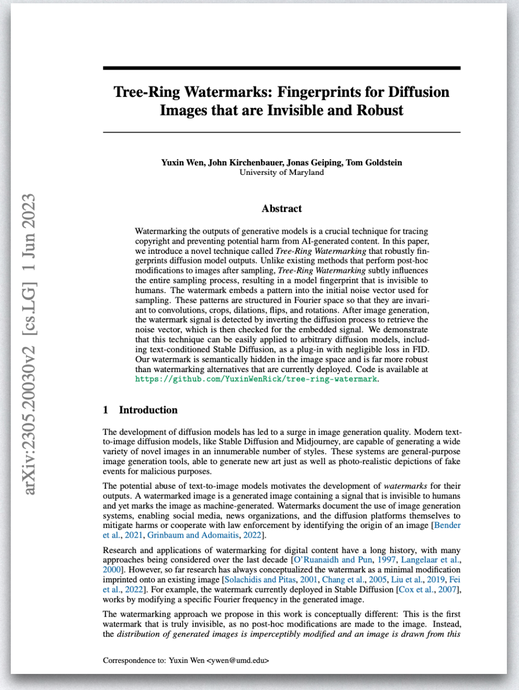
Problem. Consider a provider who wants to make their image generator publicly accessible under a contractual usage agreement that serves to prevent misuse of the model. The threat is a user who breaks this agreement and uses the generator to synthesize and distribute harmful deepfakes without detection. 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

§To appear at USENIX Security 2023.

1

USENIX'23, Lukas et al.



Tree-Ring Watermarks: Fingerprints for Diffusion Images that are Invisible and Robust

Yuxin Wen, John Kirchenbauer, Jonas Geiping, Tom Goldstein
University of Maryland

arXiv:2305.20030v2 [cs.LG] 1 Jun 2023

Abstract

Watermarking the outputs of generative models is a crucial technique for tracing copyright and preventing potential harm from AI-generated content. In this paper, we introduce a novel technique called *Tree-Ring Watermarking* that robustly fingerprints diffusion model outputs. Unlike existing methods that perform post-hoc modifications to images after sampling, *Tree-Ring Watermarking* subtly influences the entire sampling process, resulting in a model fingerprint that is invisible to humans. The watermark embeds a pattern into the initial noise vector used for sampling. These patterns are structured in Fourier space so that they are invariant to convolutions, crops, dilations, flips, and rotations. After image generation, the watermark signal is detected by inverting the diffusion process to retrieve the noise vector, which is then checked for the embedded signal. We demonstrate that this technique can be easily applied to arbitrary diffusion models, including text-conditioned Stable Diffusion, as a plug-in with negligible loss in FID. Our watermark is semantically hidden in the image space and is far more robust than watermarking alternatives that are currently deployed. Code is available at https://github.com/YuxinWenRick/tree-ring-watermark.

1 Introduction

The development of diffusion models has led to a surge in image generation quality. Modern text-to-image diffusion models, like Stable Diffusion and Midjourney, are capable of generating a wide variety of novel images in an innumerable number of styles. These systems are general-purpose image generation tools, able to generate new art just as well as photo-realistic depictions of fake events for malicious purposes.

The potential abuse of text-to-image models motivates the development of *watermarks* for their outputs. A watermarked image is a generated image containing a signal that is invisible to humans and yet marks the image as machine-generated. Watermarks document the use of image generation systems, enabling social media, news organizations, and the diffusion platforms themselves to mitigate harms or cooperate with law enforcement by identifying the origin of an image [Bender et al., 2021, Grinbaum and Adomaitis, 2022].

Research and applications of watermarking for digital content have a long history, with many approaches being considered over the last decade [O'Ruanaidh and Pun, 1997, Langelaar et al., 2000]. However, so far research has always conceptualized the watermark as a minimal modification imprinted onto an existing image [Solachidis and Pitas, 2001, Chang et al., 2005, Liu et al., 2019, Fei et al., 2022]. For example, the watermark currently deployed in Stable Diffusion [Cox et al., 2007], works by modifying a specific Fourier frequency in the generated image.

The watermarking approach we propose in this work is conceptually different: This is the first watermark that is truly invisible, as no post-hoc modifications are made to the image. Instead, the *distribution of generated images is imperceptibly modified and an image is drawn from this*

Correspondence to: Yuxin Wen <ywen@umd.edu>

Preprint, Wen et al.

Related Work

Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data

Ning Yu[1,2*] Vladislav Skripniuk[3*] Sahar Abdelnabi[3] Mario Fritz[3]
[1]University of Maryland [2]Max Planck Institute for Informatics
[3]CISPA Helmholtz Center for Information Security
{ningyu,vladislav}@mpi-inf.mpg.de {sahar.abdelnabi,fritz}@cispa.saarland

Abstract

Photorealistic image generation has reached a new level of quality due to the breakthroughs of generative adversarial networks (GANs). Yet, the dark side of such deepfakes, the malicious use of generated media, raises concerns about visual misinformation. While existing research work on deepfake detection demonstrates high accuracy, it is subject to advances in generation techniques and adversarial iterations on detection countermeasure techniques. Thus, we seek a proactive and sustainable solution on deepfake detection, that is agnostic to the evolution of generative models, by introducing artificial fingerprints into the models.

Our approach is simple and effective. We first embed artificial fingerprints into training data, then validate a surprising discovery on the transferability of such fingerprints from training data to generative models, which in turn appears in the generated deepfakes. Experiments show that our fingerprinting solution (1) holds for a variety of cutting-edge generative models, (2) leads to a negligible side effect on generation quality, (3) stays robust against image-level and model-level perturbations, (4) stays hard to be detected by adversaries, and (5) converts deepfake detection and attribution into trivial tasks and outperforms the recent state-of-the-art baselines. Our solution closes the responsibility loop between publishing pre-trained generative model inventions and their possible misuses, which makes it independent of the current arms race. Code and models are available at GitHub.

1. Introduction

In the past years, photorealistic image generation has been rapidly evolving, benefiting from the invention of generative adversarial networks (GANs) [17] and its successive breakthroughs [40, 18, 36, 5, 26, 27, 28, 54, 55]. Given the level of realism and diversity that generative models can achieve today, detecting generated media, well known as *deepfakes*, attributing their sources, and tracing their legal responsibilities become infeasible to human beings.

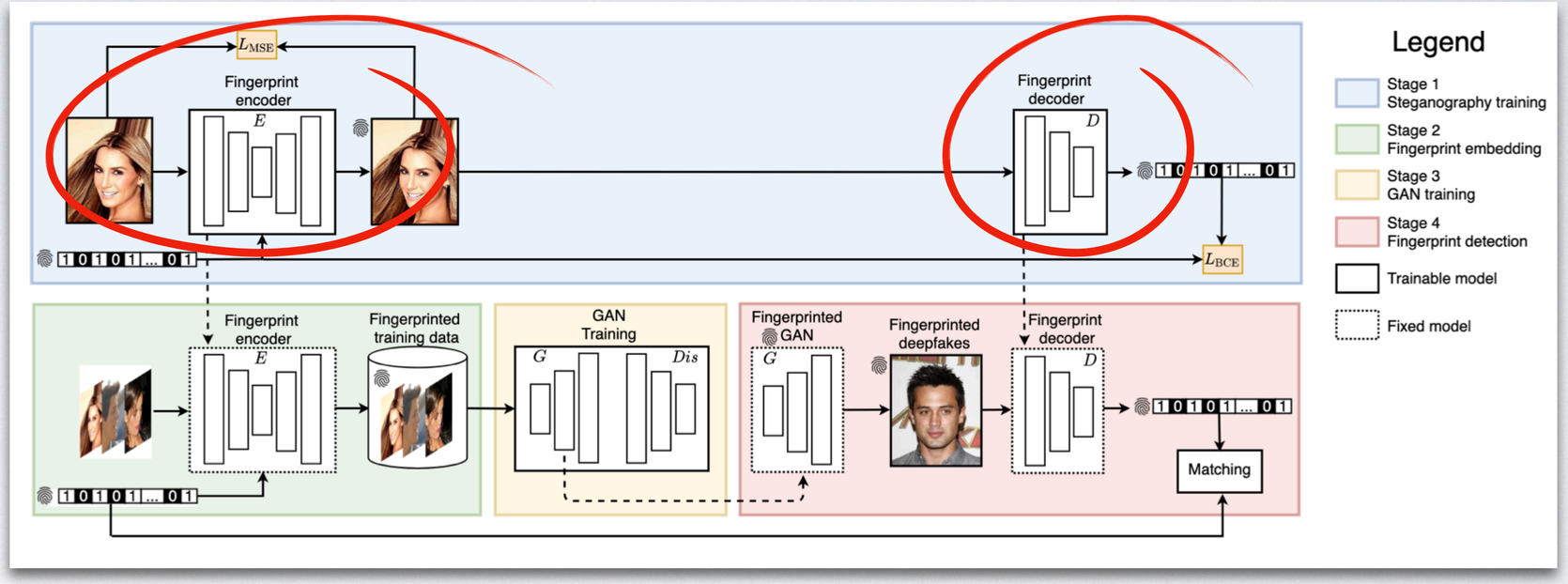
Moreover, the misuse of deepfakes has been permeating to each corner of social media, ranging from misinformation of political campaigns [25] to fake journalism [48, 42]. This motivates tremendous research efforts on deepfake detection [57] and source attribution [35, 53, 50]. These techniques aim to counter the widespread of malicious applications of deepfakes by automatically identifying and flagging generated visual contents and tracking their sources. Most of them rely on low-level visual patterns in GAN-generated images [35, 53, 50, 20, 61] or frequency mismatch [14, 60, 15]. However, these techniques are unable to sustainably and robustly prevent deepfake misuse in the long run; as generative models evolve, they learn to better match the true distribution causing fewer artifacts [57]. Besides, detection countermeasures are also continuously evolving [13, 7, 57].

Motivated by this, we tackle deepfake detection and attribution through a different lens, and propose a ***proactive*** and sustainable solution for detection, which is simple and effective. In specific, we aim to introduce ***artificial fingerprints*** into generative models that enable identification and tracing. Figure 1 depicts our pipeline; we first embed artificial fingerprints into the training data using image steganography [4, 45]. The generative model is then trained with its original protocol without modification. This makes our solution agnostic and plug-and-play for arbitrary models. We then show a surprising discovery on the ***transferability*** of such fingerprints from training data to the model: the same fingerprint information that was encoded in the training data can be decoded from all generated images.

We achieve deepfake detection by classifying images with matched fingerprints in our database as fake and images with random detected fingerprints as real. We also achieve deepfake attribution when we allocate different fingerprints for different generative models. Our solution thus closes the responsibility loop between generative model in-

*Equal contribution.

ICCV'21, Yu et al.



- Key generation is model-independent
- Scalability: requires re-training from scratch
- Poor effectiveness
- No adaptive attacker

Related Work

Published as a conference paper at ICLR 2022

RESPONSIBLE DISCLOSURE OF GENERATIVE MODELS USING SCALABLE FINGERPRINTING

Ning Yu[1,2,3†*] Vladislav Skripniuk[4†] Dingfan Chen[4] Larry Davis[2] Mario Fritz[4]
[1]Salesforce Research [2]University of Maryland [3]Max Planck Institute for Informatics
[4]CISPA Helmholtz Center for Information Security
ning.yu@salesforce.com vladislav@mpi-inf.mpg.de
{dingfan.chen, fritz}@cispa.de lsd@cs.umd.edu

arXiv:2012.08726v5 [cs.CR] 17 Mar 2022

ABSTRACT

Over the past years, deep generative models have achieved a new level of performance. Generated data has become difficult, if not impossible, to be distinguished from real data. While there are plenty of use cases that benefit from this technology, there are also strong concerns on how this new technology can be misused to generate deep fakes and enable misinformation at scale. Unfortunately, current deep fake detection methods are not sustainable, as the gap between real and fake continues to close. In contrast, our work enables a responsible disclosure of such state-of-the-art generative models, that allows model inventors to fingerprint their models, so that the generated samples containing a fingerprint can be accurately detected and attributed to a source. Our technique achieves this by an efficient and scalable ad-hoc generation of a large population of models with distinct fingerprints. Our recommended operation point uses a 128-bit fingerprint which in principle results in more than 10^{38} identifiable models. Experiments show that our method fulfills key properties of a fingerprinting mechanism and achieves effectiveness in deep fake detection and attribution. Code and models are available at GitHub.

1 INTRODUCTION

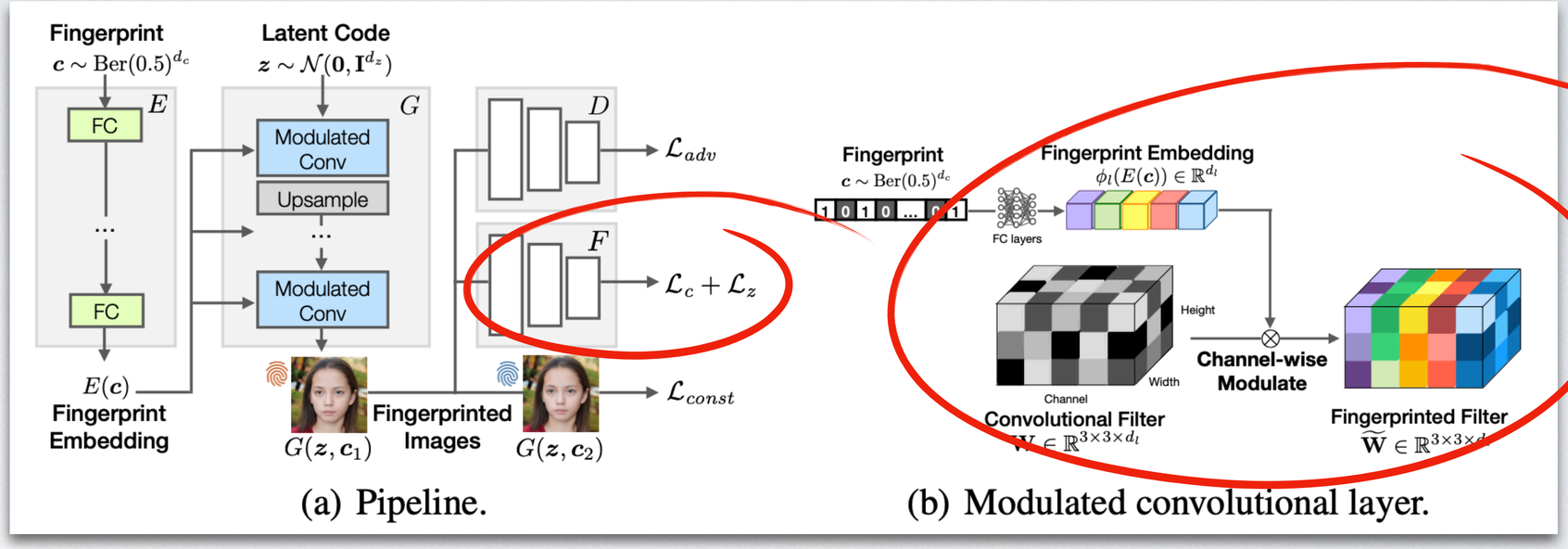
Over the recent eight years, deep generative models have demonstrated stunning performance in generating photorealistic images, considerably boosted by the revolutionary technique of generative adversarial networks (GANs) (Goodfellow et al., 2014; Radford et al., 2016; Gulrajani et al., 2017; Miyato et al., 2018; Brock et al., 2018; Karras et al., 2018; 2019; 2020).

However, with the closing margins between real and fake, a flood of strong concerns arise (Harris, 2018; Chesney & Citron, 2019; Brundage et al., 2018): how if these models are misused to spoof sensors, generate deep fakes, and enable misinformation at scale? Not only human beings have difficulties in distinguishing deep fakes, but dedicated research efforts on deep fake detection (Durall et al., 2019; Zhang et al., 2019; Frank et al., 2020; Zhang et al., 2020) and attribution (Marra et al., 2019; Yu et al., 2019; Wang et al., 2020) are also unable to sustain longer against the evolution of generative models. For example, researchers delve into details on how deep fake detection works, and learn to improve generation that better fits the detection criteria (Zhang et al., 2020; Durall et al., 2020). In principle, any successful detector can play an auxiliary role in augmenting the discriminator in the next iteration of GAN techniques, and consequently results in an even stronger generator.

The dark side of deep generative models delays its industrialization process. For example, when commercializing the GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) models, OpenAI leans conservative to open-source their models but rather only release the black-box APIs[1]. They involve expensive human labor in the loop to review user downloads and monitor the usage of their APIs. Yet still, it is a challenging and industry-wide task on how to trace the responsibility of the downstream use cases in an open end.

To pioneer in this task, we propose a model fingerprinting mechanism that enables responsible release and regulation of generative models. In particular, we allow responsible model inventors to

*This work was done when Ning Yu was in a joint Ph.D. program with the University of Maryland and Max Planck Institute for Informatics.
†Equal contribution.
[1]https://openai.com/blog/openai-api/



(a) Pipeline. (b) Modulated convolutional layer.

ICLR'22, Yu et al.

+ Key generation is model-dependent

- Scalability: Key generation requires re-training and is unstable

- Poor effectiveness

- No adaptive attacker

Related Work

arXiv:2304.07361v2 [cs.LG] 24 May 2023

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

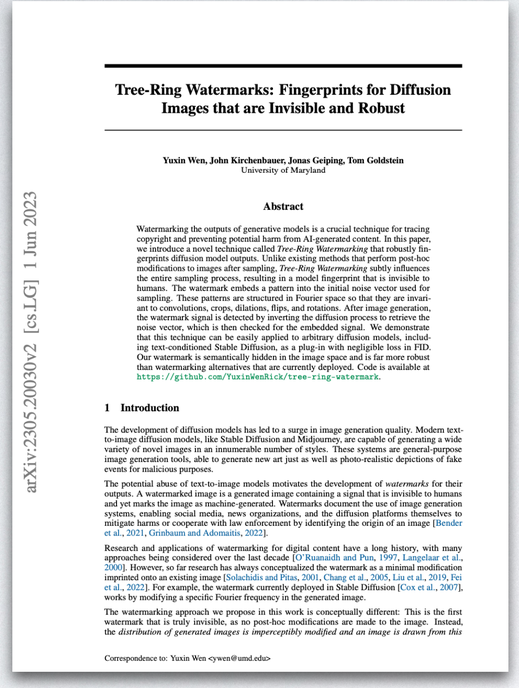
[1]To appear at USENIX Security 2023.

1

USENIX'23, Lukas et al.

We will show this later …

Related Work



arXiv:2305.20030v2 [cs.LG] 1 Jun 2023

Tree-Ring Watermarks: Fingerprints for Diffusion Images that are Invisible and Robust

Yuxin Wen, John Kirchenbauer, Jonas Geiping, Tom Goldstein
University of Maryland

Abstract

Watermarking the outputs of generative models is a crucial technique for tracing copyright and preventing potential harm from AI-generated content. In this paper, we introduce a novel technique called *Tree-Ring Watermarking* that robustly fingerprints diffusion model outputs. Unlike existing methods that perform post-hoc modifications to images after sampling, *Tree-Ring Watermarking* subtly influences the entire sampling process, resulting in a model fingerprint that is invisible to humans. The watermark embeds a pattern into the initial noise vector used for sampling. These patterns are structured in Fourier space so that they are invariant to convolutions, crops, dilations, flips, and rotations. After image generation, the watermark signal is detected by inverting the diffusion process to retrieve the noise vector, which is then checked for the embedded signal. We demonstrate that this technique can be easily applied to arbitrary diffusion models, including text-conditioned Stable Diffusion, as a plug-in with negligible loss in FID. Our watermark is semantically hidden in the image space and is far more robust than watermarking alternatives that are currently deployed. Code is available at https://github.com/YuxinWenRick/tree-ring-watermark.

1 Introduction

The development of diffusion models has led to a surge in image generation quality. Modern text-to-image diffusion models, like Stable Diffusion and Midjourney, are capable of generating a wide variety of novel images in an innumerable number of styles. These systems are general-purpose image generation tools, able to generate new art just as well as photo-realistic depictions of fake events for malicious purposes.

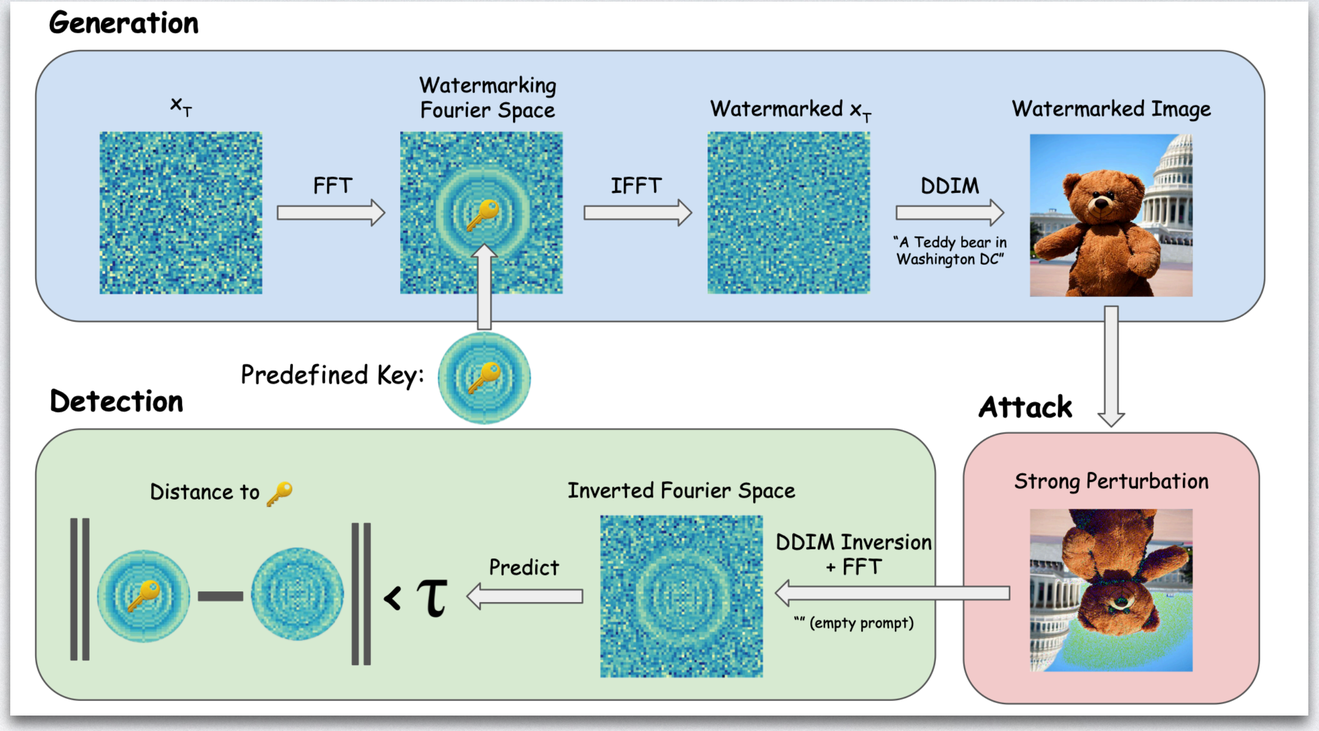
The potential abuse of text-to-image models motivates the development of *watermarks* for their outputs. A watermarked image is a generated image containing a signal that is invisible to humans and yet marks the image as machine-generated. Watermarks document the use of image generation systems, enabling social media, news organizations, and the diffusion platforms themselves to mitigate harms or cooperate with law enforcement by identifying the origin of an image [Bender et al., 2021, Grinbaum and Adomaitis, 2022].

Research and applications of watermarking for digital content have a long history, with many approaches being considered over the last decade [O'Ruanaidh and Pun, 1997, Langelaar et al., 2000]. However, so far research has always conceptualized the watermark as a mininal modification imprinted onto an existing image [Solachidis and Pitas, 2001, Chang et al., 2005, Liu et al., 2019, Fei et al., 2022]. For example, the watermark currently deployed in Stable Diffusion [Cox et al., 2007], works by modifying a specific Fourier frequency in the generated image.

The watermarking approach we propose in this work is conceptually different: This is the first watermark that is truly invisible, as no post-hoc modifications are made to the image. Instead, the *distribution of generated images is imperceptibly modified and an image is drawn from this*

Correspondence to: Yuxin Wen <ywen@umd.edu>

Preprint, Wen et al.



+ Scalable, no re-training required

- Key generation is model-independent

- Requires control over the generator

- No adaptive attacker

Pivotal Tuning Watermarking

🏆 Goals

Effectiveness: Long messages at low quality degradation

Robustness against black-box and white-box attacks

Undetectability against black-box attacker

Scalability: Embed into pre-trained generators

Generative Adversarial Network

Our work evaluates watermarking for Generate Adversarial Networks (GANs)

Latent code z → Generator →

StyleGAN

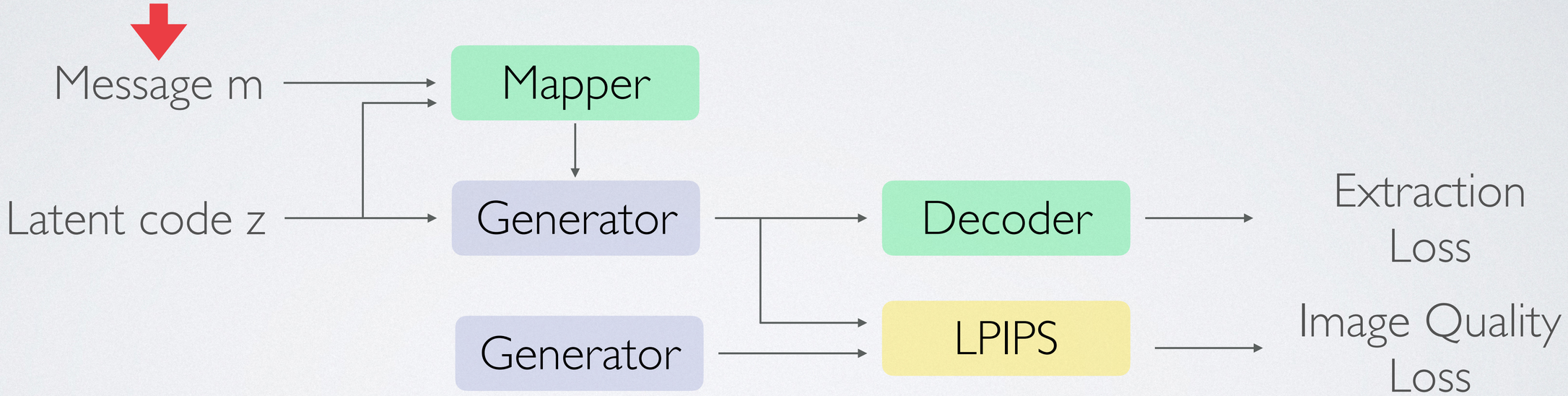
Our work evaluates watermarking for Generate Adversarial Networks (GANs)

Latent code z → Styles w ⇉ StyleGAN →

many styles to
control the output

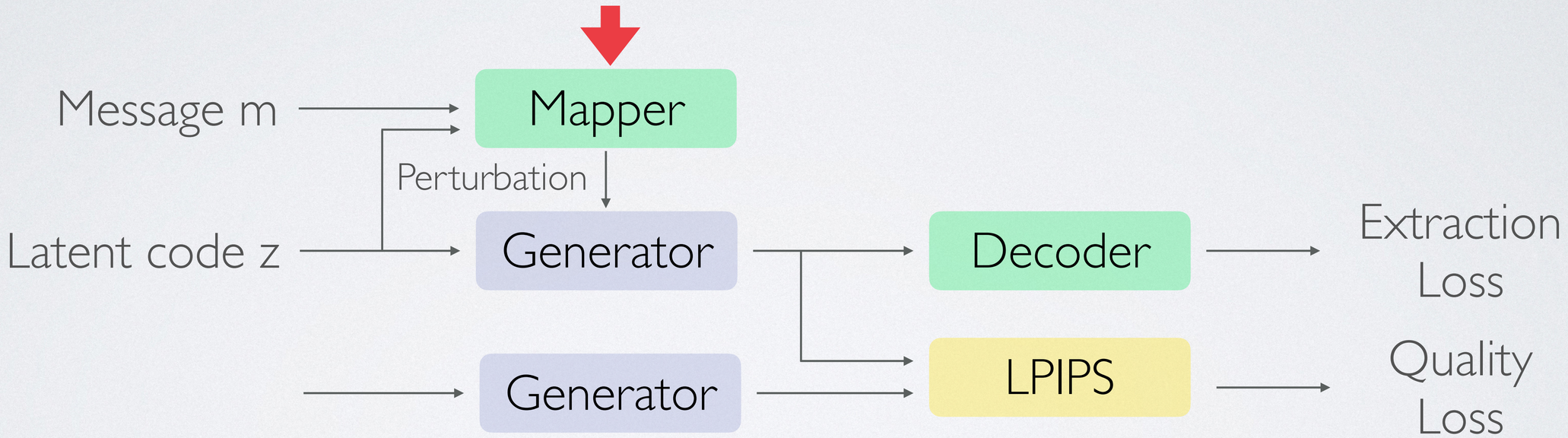
Pivotal Tuning Watermarking (1/2) - KeyGen

1) Key Generation



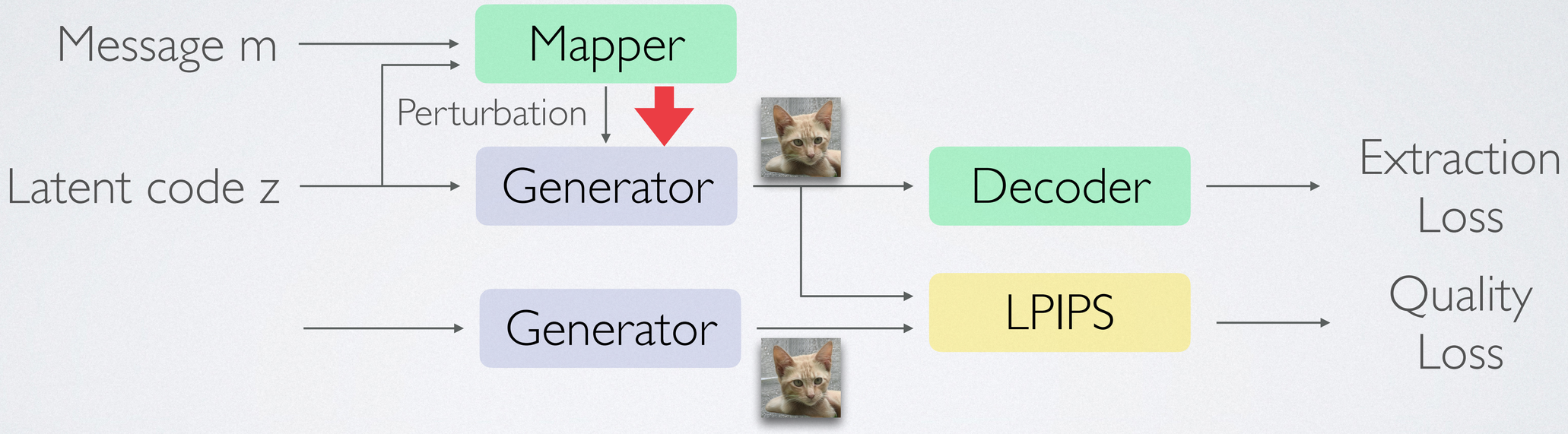
Pivotal Tuning Watermarking (1/2) - KeyGen

1) Key Generation



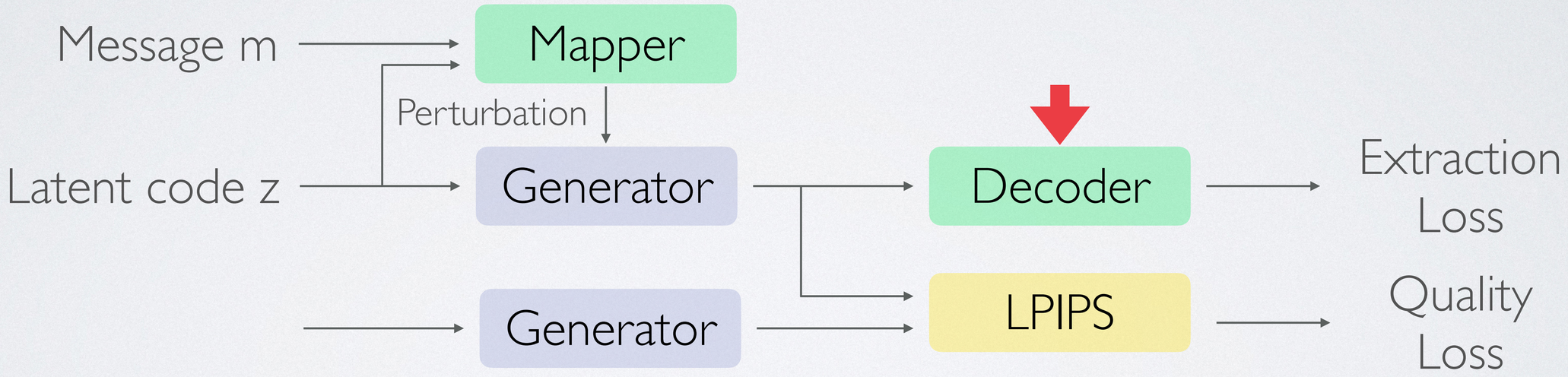
Pivotal Tuning Watermarking (1/2) - KeyGen

1) Key Generation



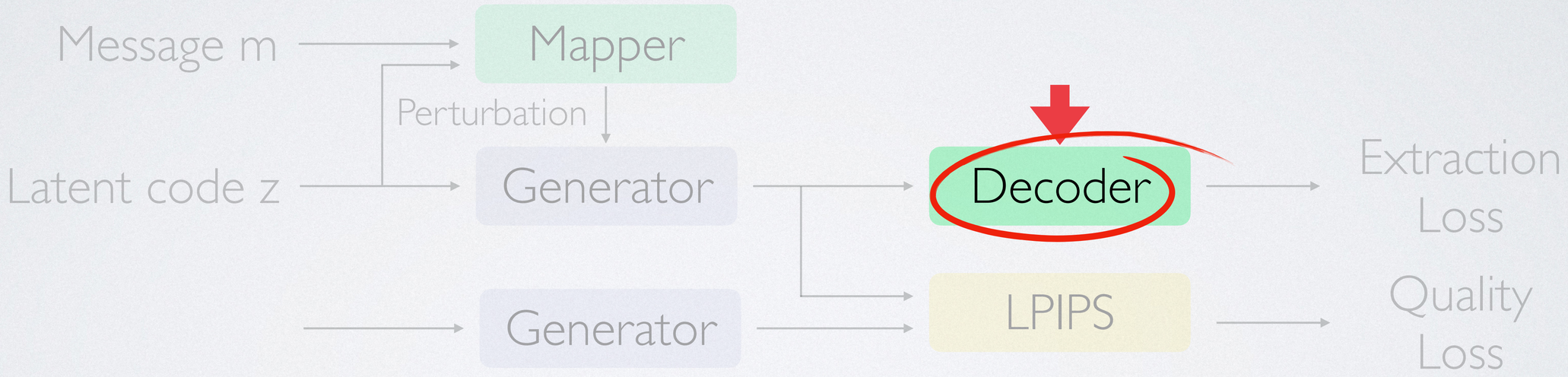
Pivotal Tuning Watermarking (1/2) - KeyGen

1) Key Generation

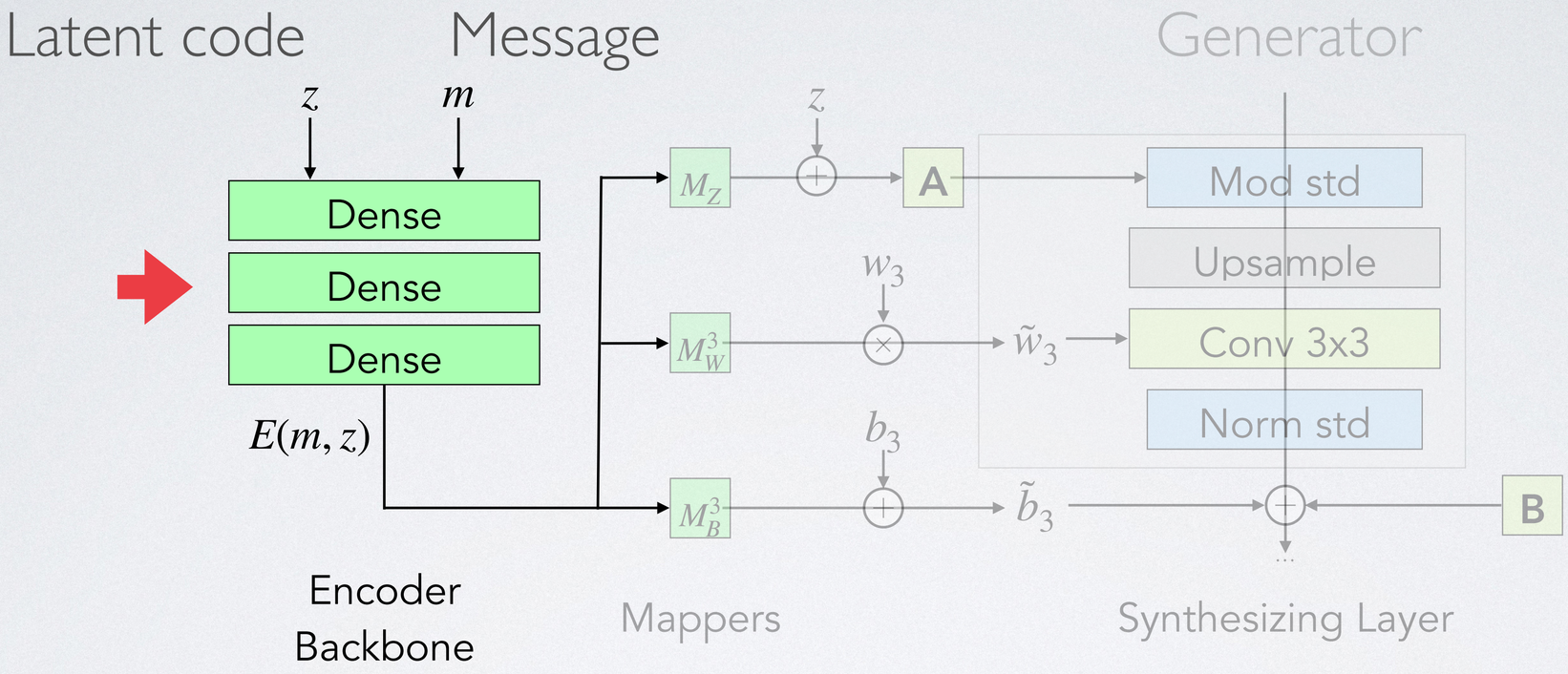


Pivotal Tuning Watermarking (1/2) - KeyGen

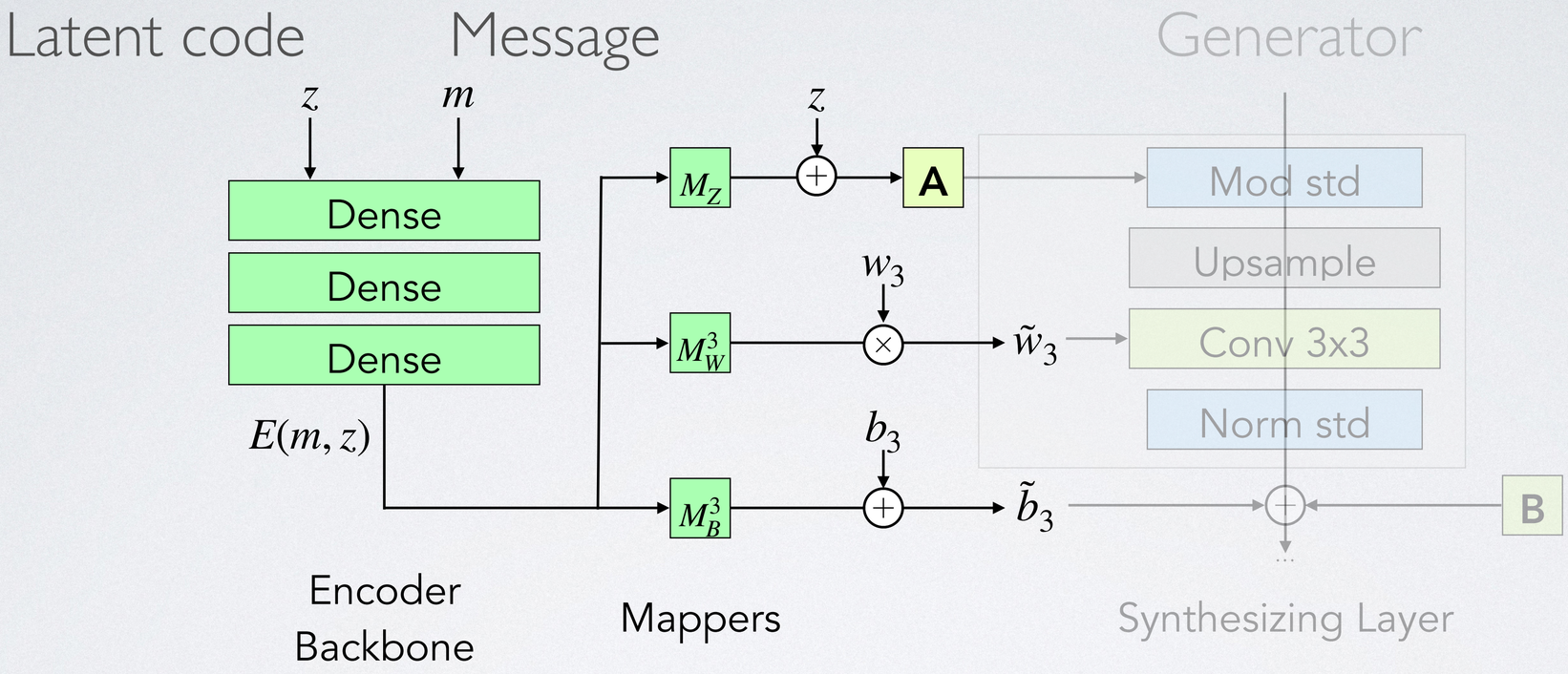
1) Key Generation



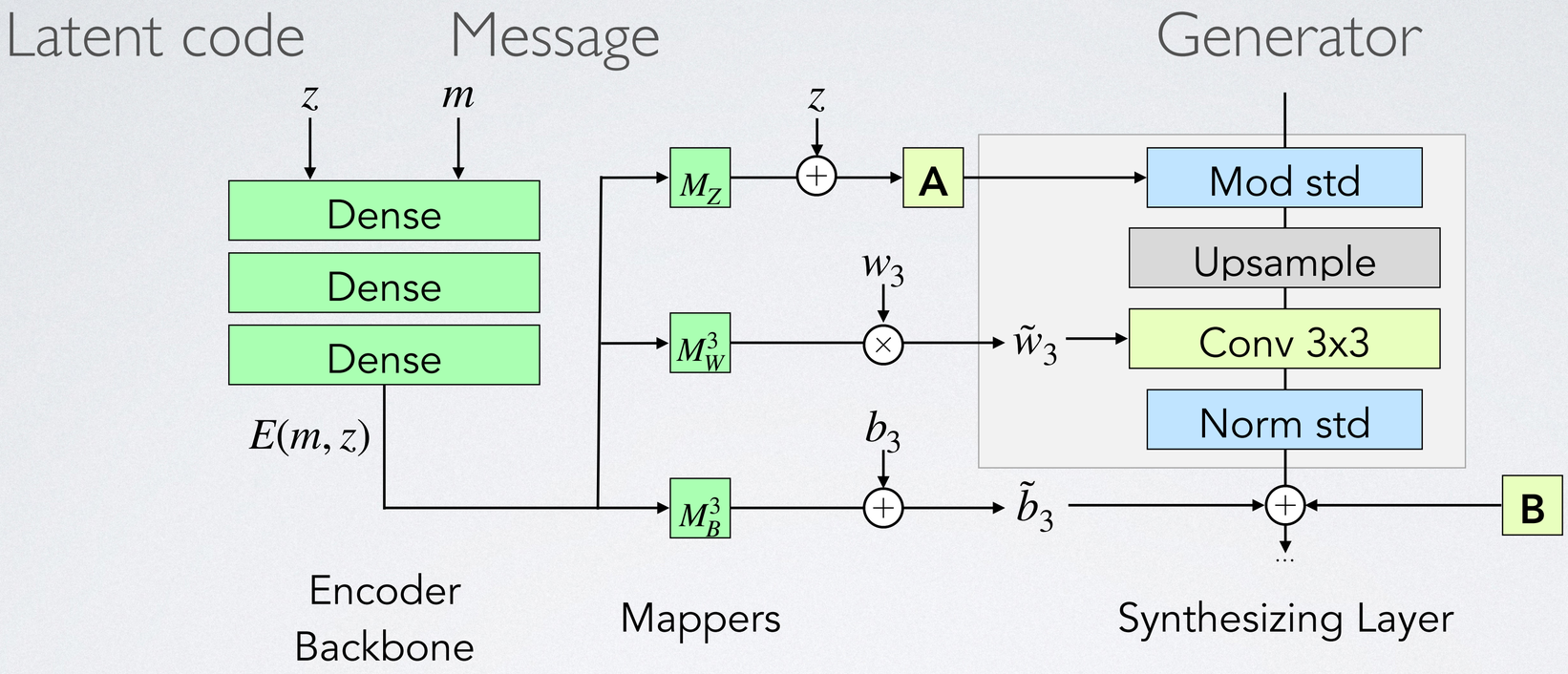
Insights into the Mapper



Insights into the Mapper



Insights into the Mapper



Pivotal Tuning Watermarking (1/2) - KeyGen

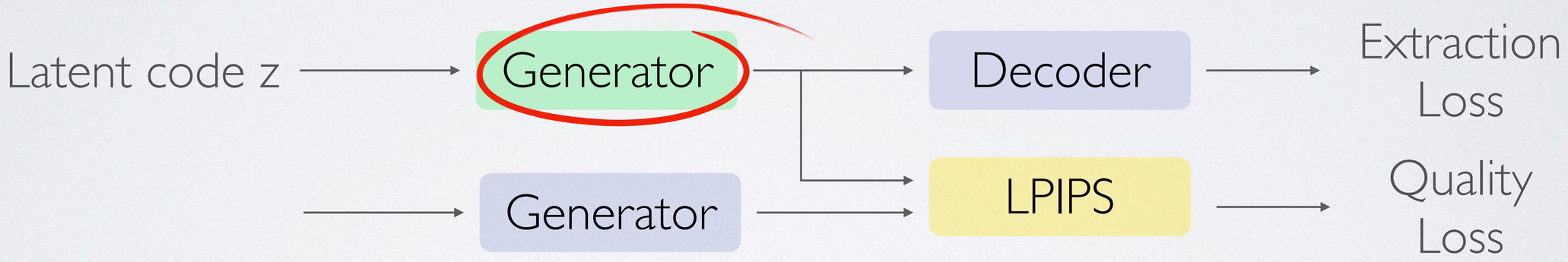
Mapper modulates inputs, weights and biases

Mapper discovers similar output distributions that are close in parameter-space

KeyGen is scalable: No real training data required & 2-4 hours on A100 GPU

Pivotal Tuning Watermarking (2/2) - Embedding

2) Pivotal Tuning



Pivotal Tuning Watermarking (2/2) - Embedding

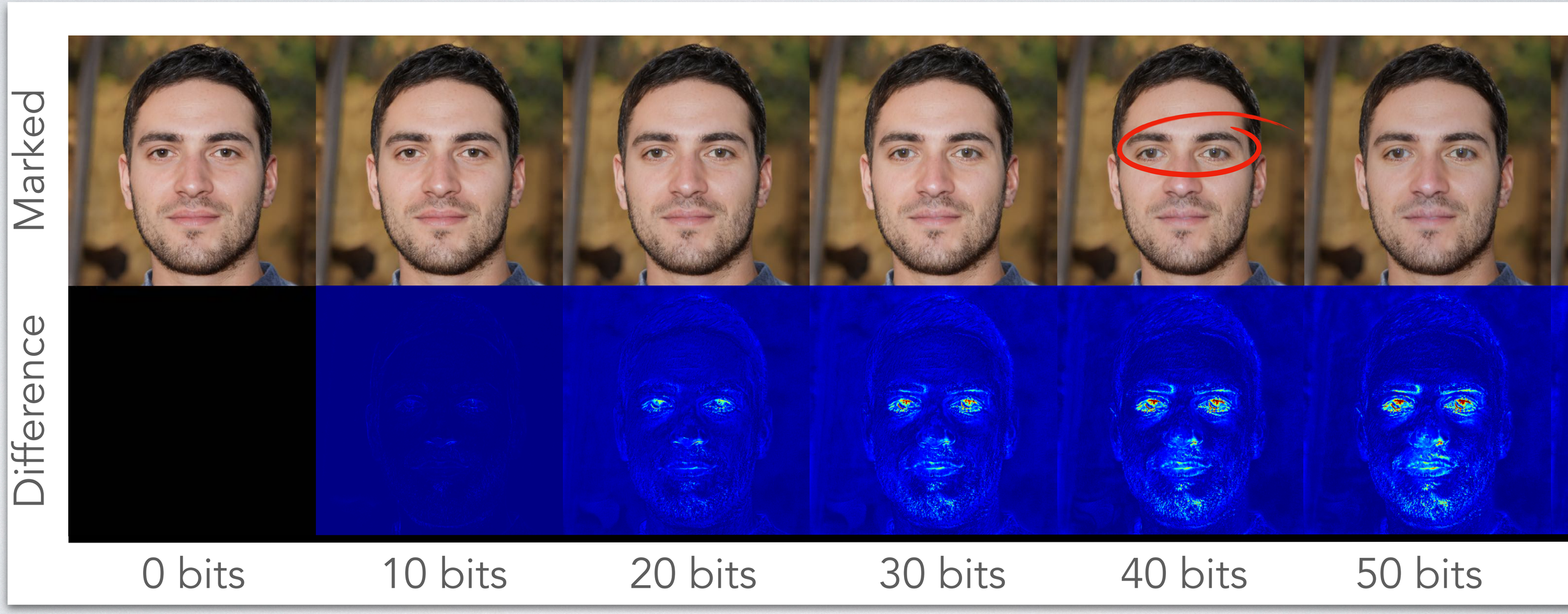
Output of PTW is a generator with watermarked parameters

Minimize distance between old and new network in the LPIPS space

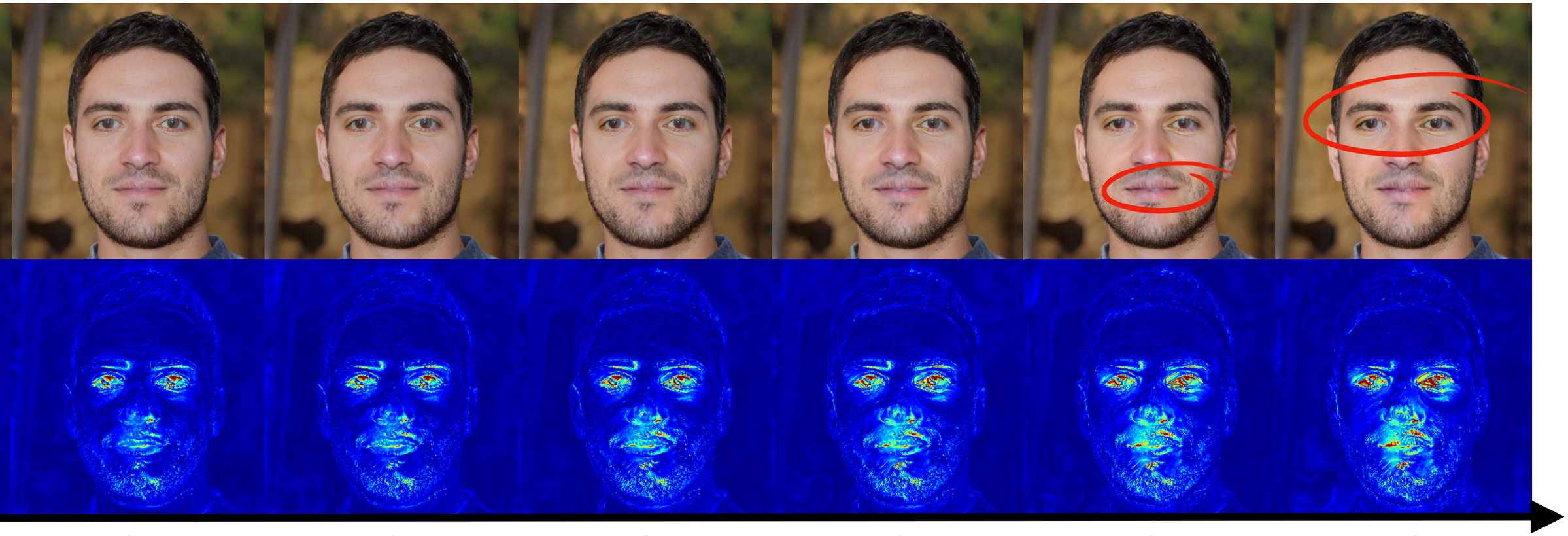
Maximize distance in the watermarking message space

Embedding is scalable: <1h on A100 GPU (~1000 times faster than training from scratch)

Effectiveness

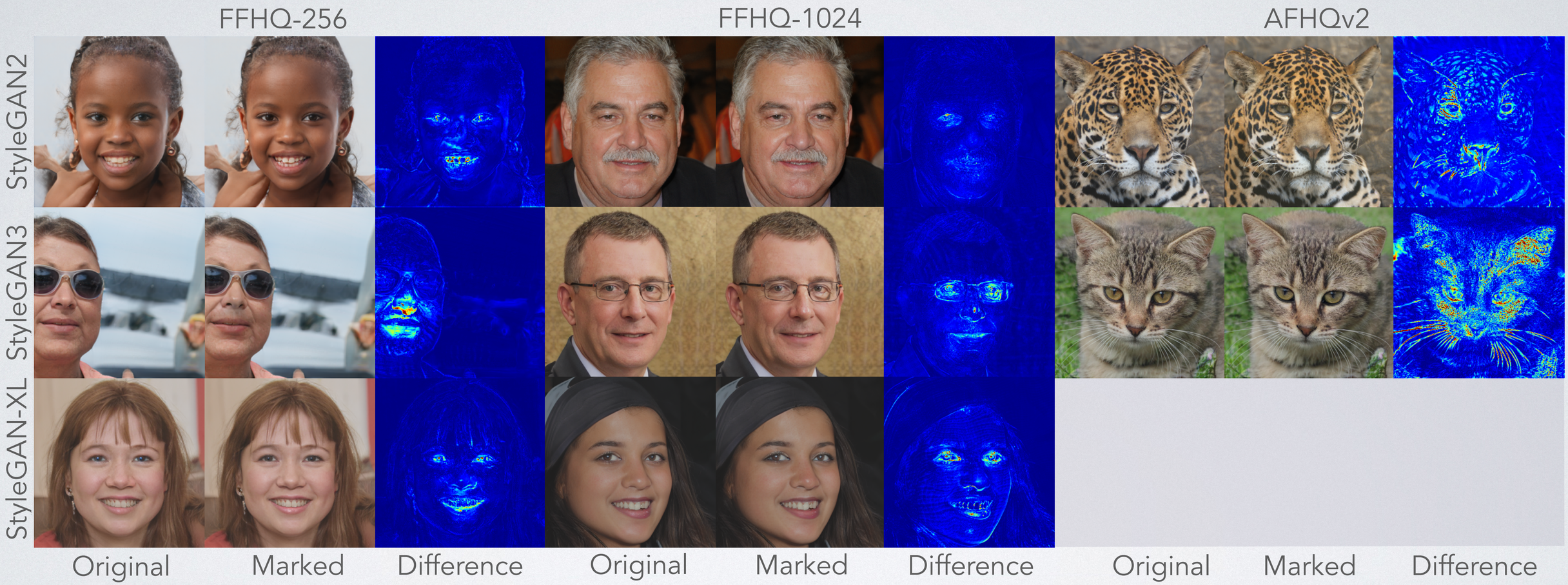


Effectiveness

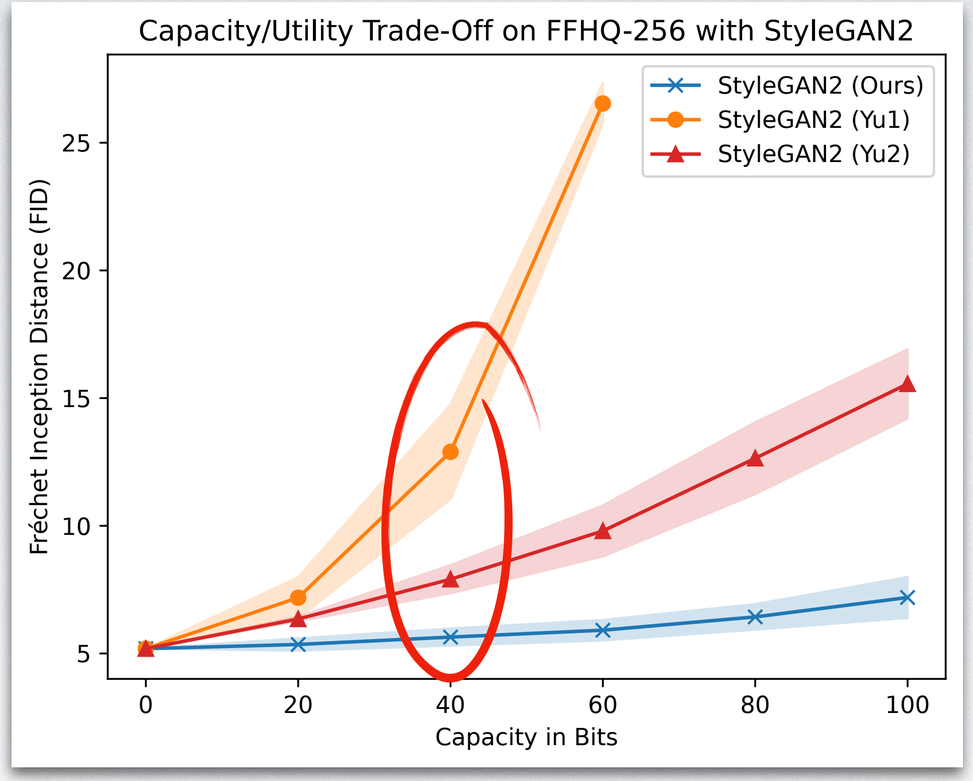


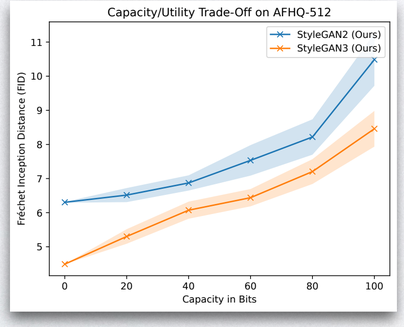
50 bits | 60 bits | 70 bits | 80 bits | 90 bits | 100 bits

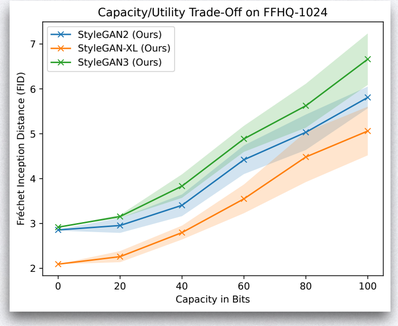
Effectiveness



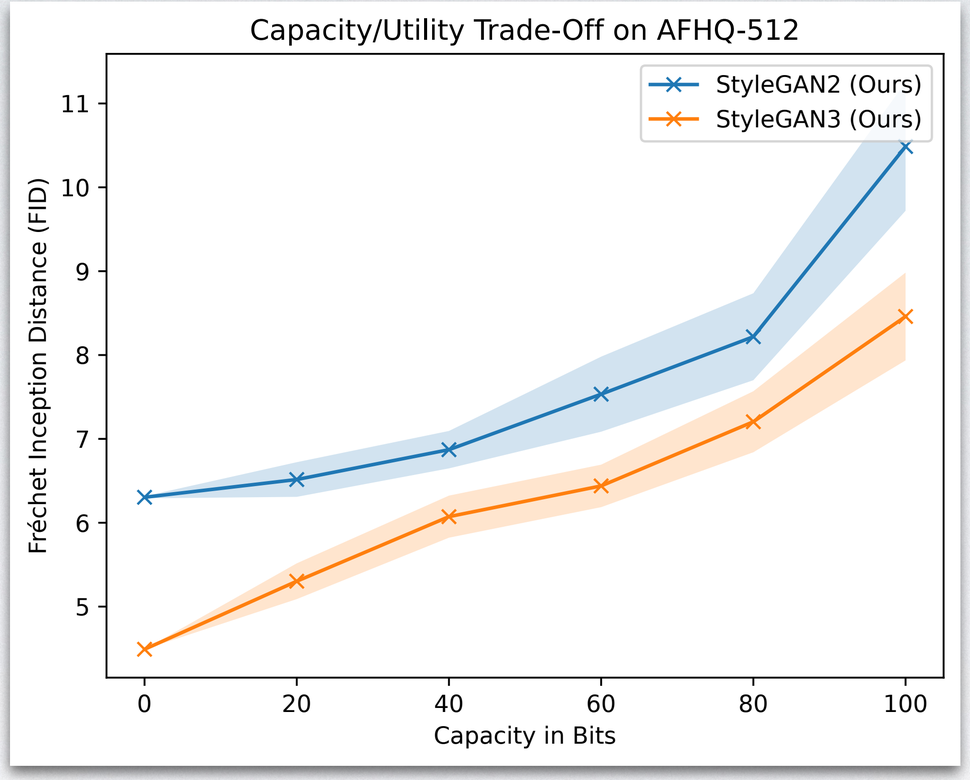
Capacity/Utility Trade-Off

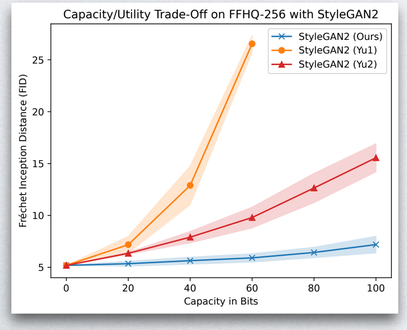


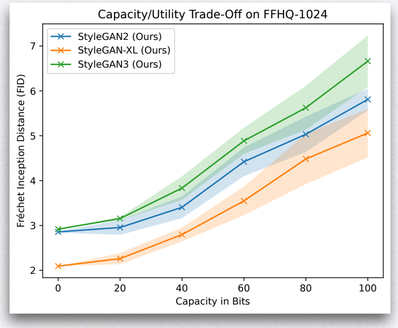




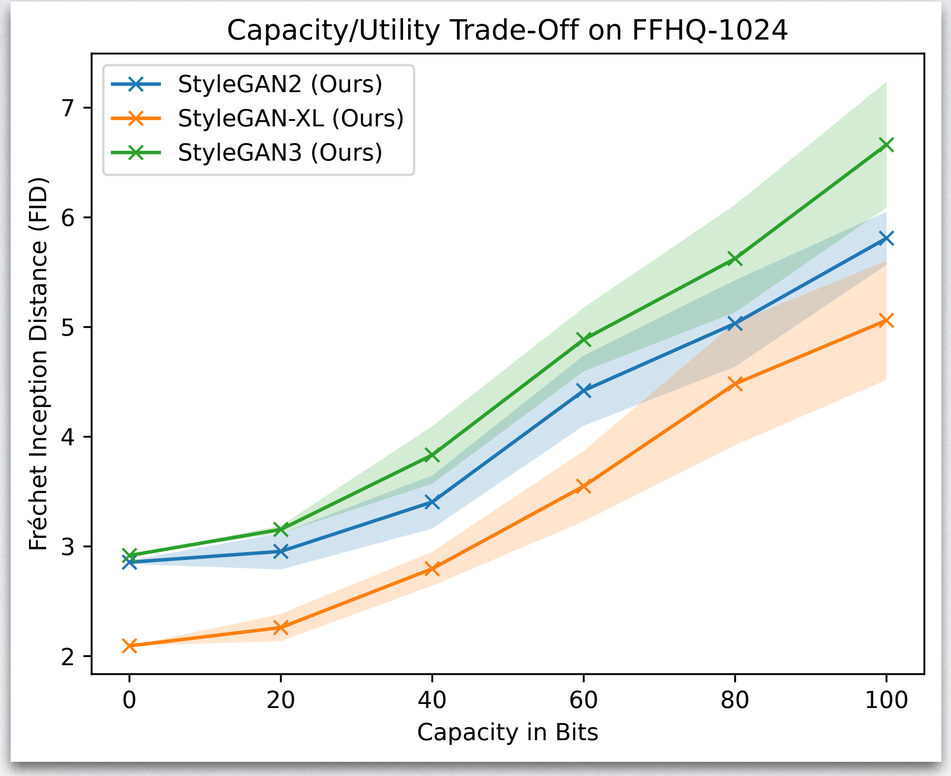
Capacity/Utility Trade-Off

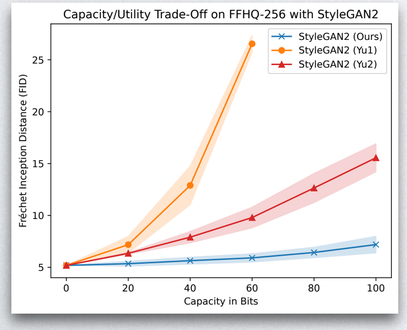


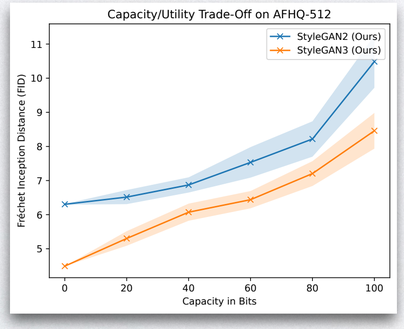




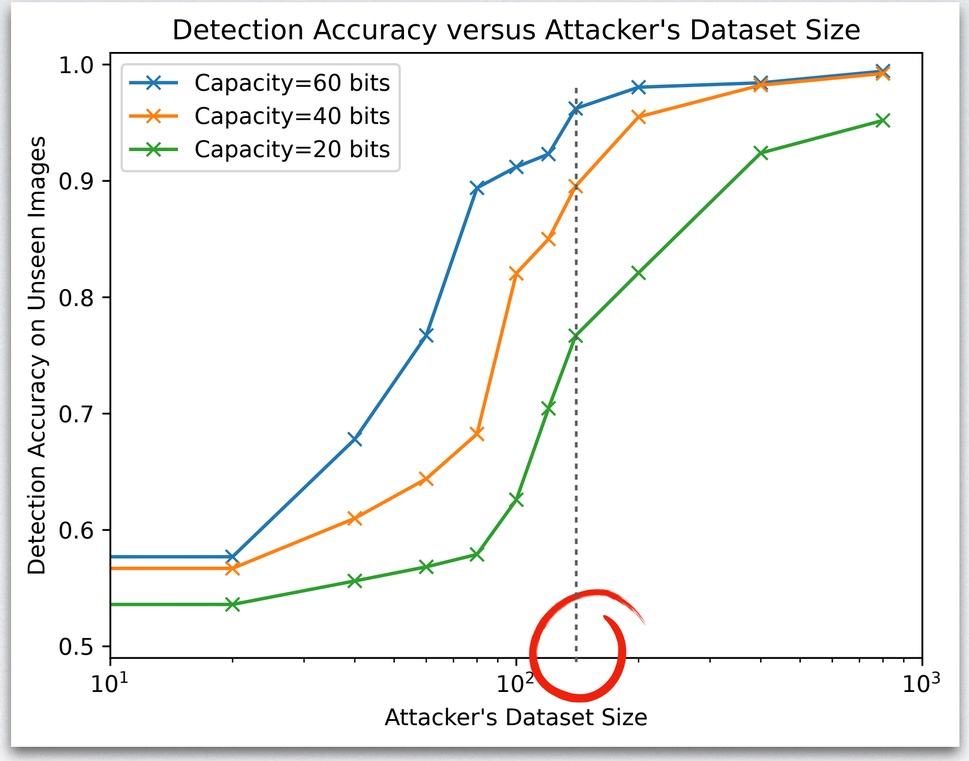
Capacity/Utility Trade-Off

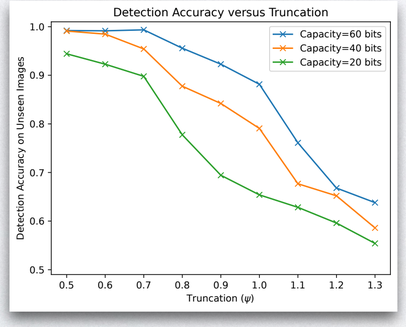






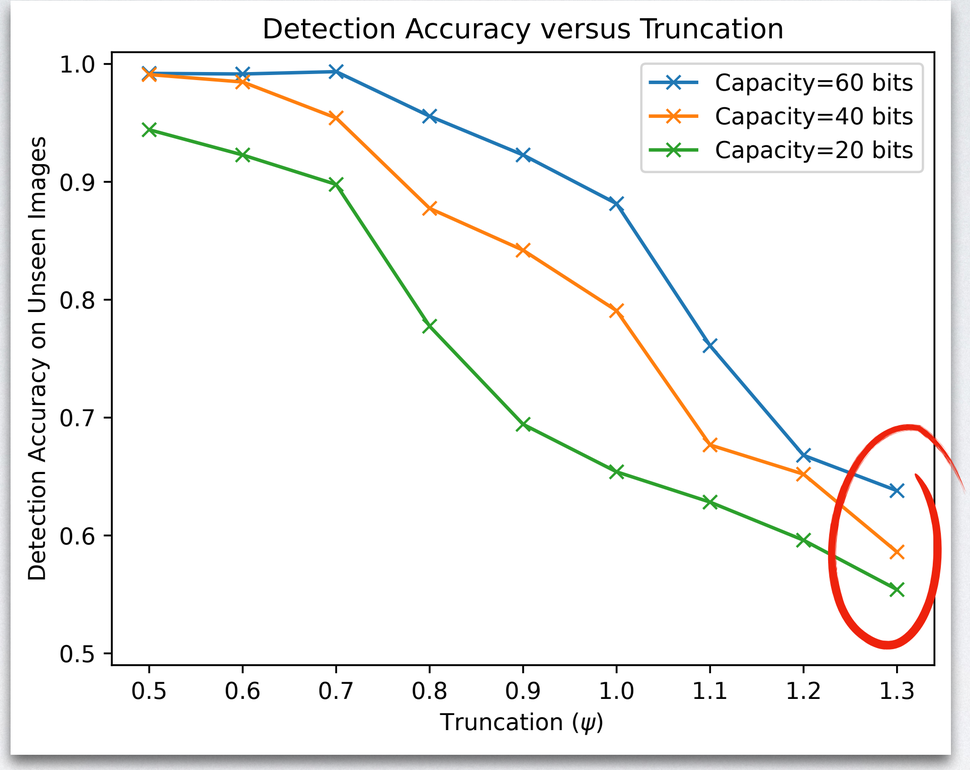
Detectability



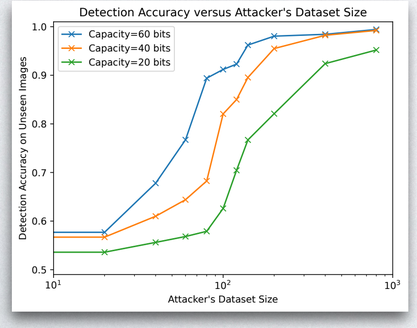


Detectability

Low variation



High variation



Variation makes detectability more difficult for the adversary

Robustness

Black-box

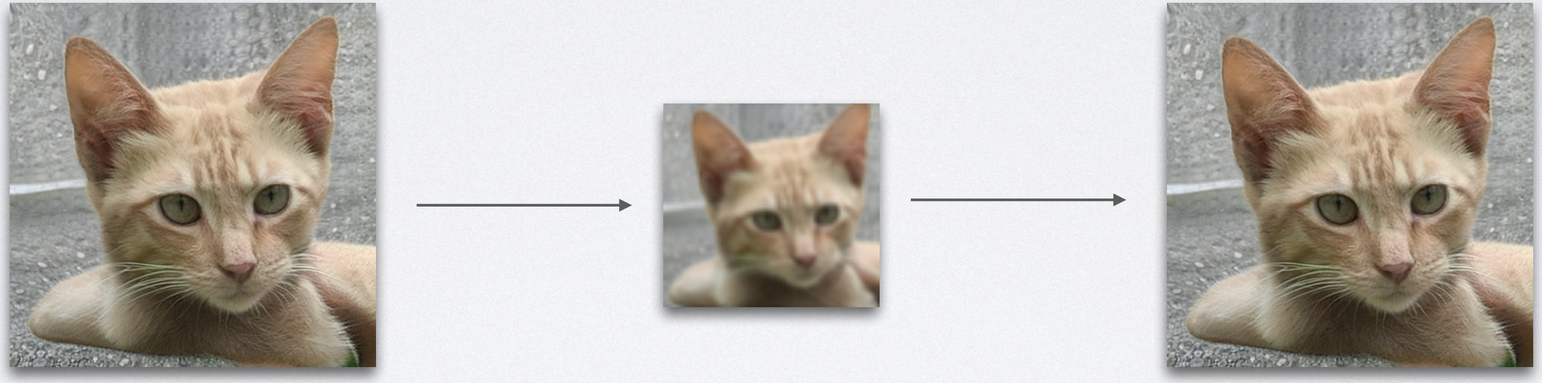
Cropping, Blurring, Quantization, Noising, JPEG Compression
+ Super-Resolution Models

White-box

Reverse Pivotal Tuning, Overwriting

Robustness (1/3) - Super-Resolution

Black-box



Generated Image → downscaling → Super-Resolution

Robustness (2/3) - Overwriting

White-box

1) Use our approach to train a surrogate decoder

2) Sample random message and overwrite parameters using PTW

Robustness (3/3) - Reverse Pivotal Tuning

White-box

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

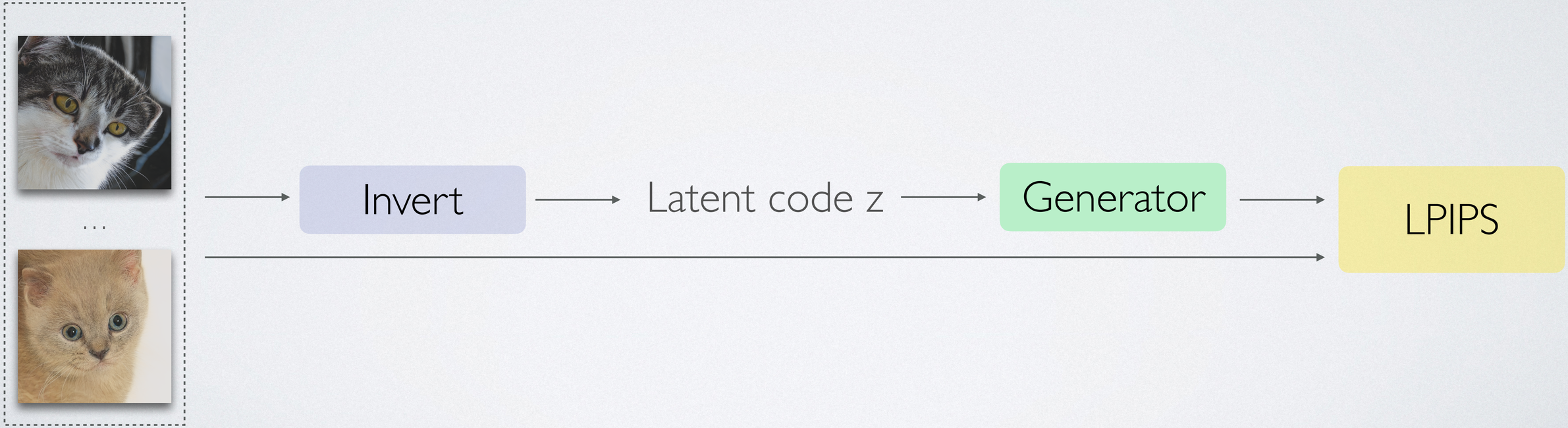
2) Regularize generator with Pivotal Tuning and LPIPS loss to synthesize real images

Robustness (3/3) - Reverse Pivotal Tuning

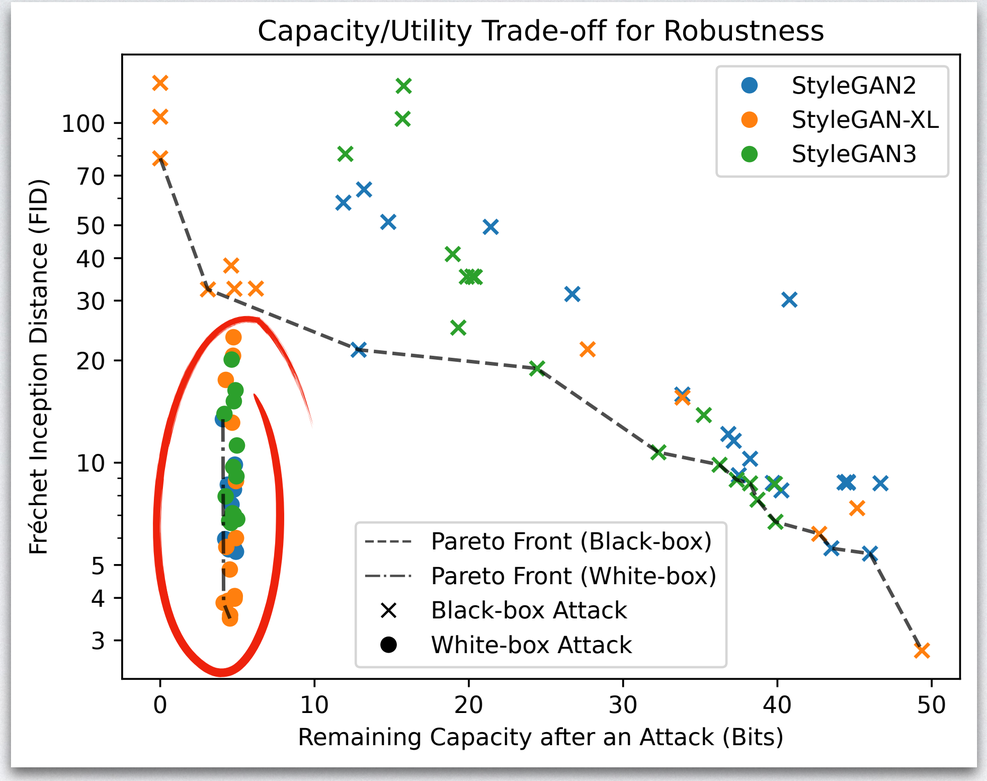
White-box

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

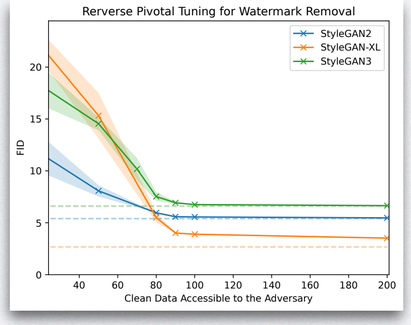
2) Regularize generator with Pivotal Tuning and LPIPS loss to synthesize real images



Robustness

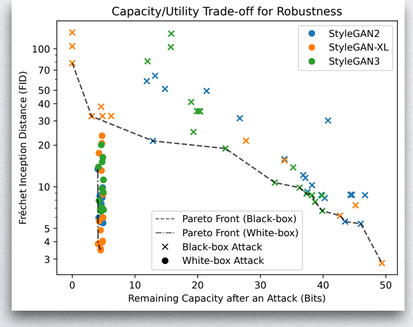


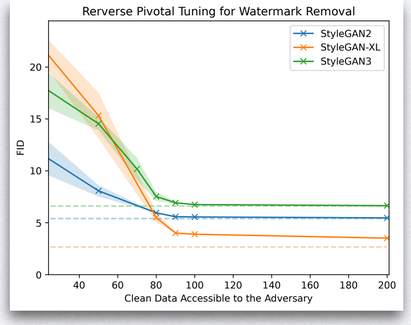
	StyleGAN2		StyleGAN-XL		StyleGAN3	
	C_θ	FID	C_θ	FID	C_θ	FID
Attacks	43.05	5.4	48.79	2.67	40.33	6.61
Black-box Attacks						
Crop	39.73	8.72	42.71	6.18	38.23	8.69
Blur	38.82	36.84	12.12	10.32	35.12	11.73
JPEG	42.12	8.70	38.43	9.12	38.23	9.33
Noise	40.26	8.29	45.17	7.35	32.29	10.73
Quantize	37.17	11.60	43.27	5.61	39.72	8.71
SR	32.86	11.51	34.52	11.62	30.12	11.34
White-box Attacks						
Overwrite	4.78	8.34	4.91	8.83	4.73	9.71
RPT_{200}	4.91	5.47	4.52	3.52	4.59	6.65
RPT_{100}	4.44	5.56	4.21	3.90	4.47	6.75
RPT_{50}	4.38	8.07	4.38	15.32	4.16	14.47



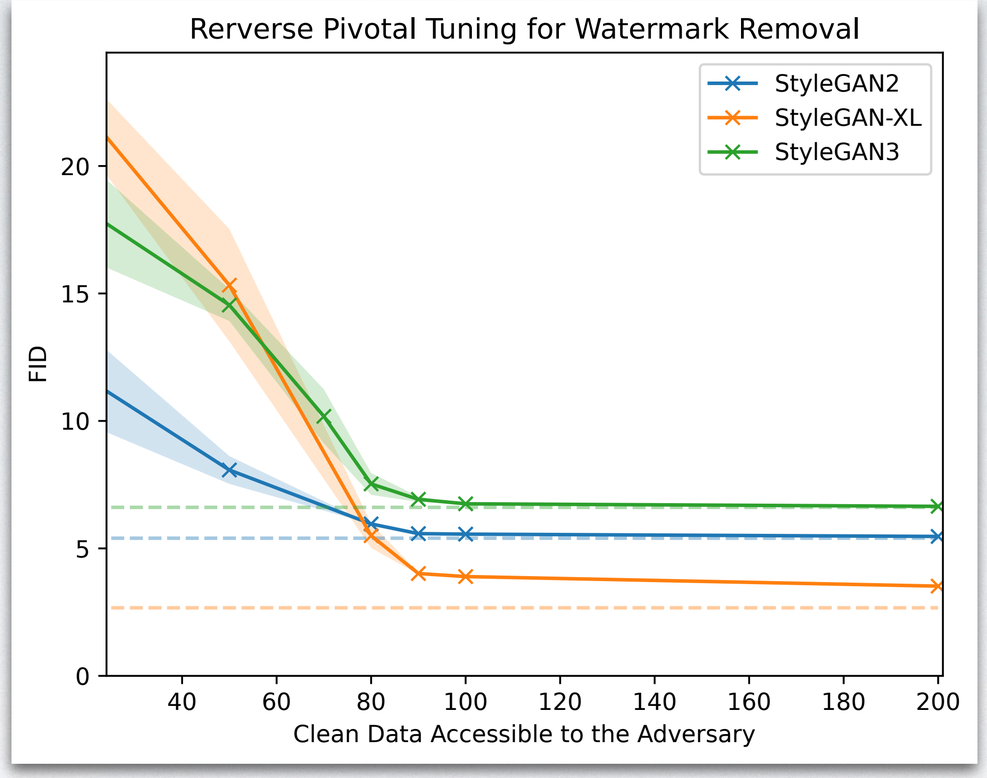
Robustness

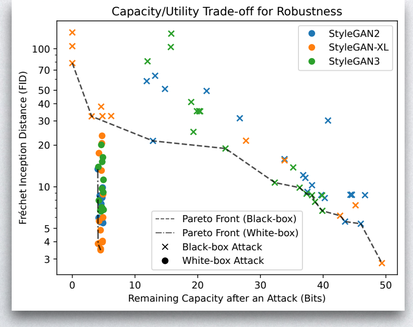
	StyleGAN2		StyleGAN-XL		StyleGAN3	
	C_θ	FID	C_θ	FID	C_θ	FID
Attacks	43.05	5.4	48.79	2.67	40.33	6.61
Black-box Attacks						
Crop	39.73	8.72	42.71	6.18	38.23	8.69
Blur	38.82	36.84	12.12	10.32	35.12	11.73
JPEG	42.12	8.70	38.43	9.12	38.23	9.33
Noise	40.26	8.29	45.17	7.35	32.29	10.73
Quantize	37.17	11.60	43.27	5.61	39.72	8.71
SR	32.86	11.51	34.52	11.62	30.12	11.34
White-box Attacks						
Overwrite	4.78	8.34	4.91	8.83	4.73	9.71
RPT_{200}	4.91	5.47	4.52	3.52	4.59	6.65
RPT_{100}	4.44	5.56	4.21	3.90	4.47	6.75
RPT_{50}	4.38	8.07	4.38	15.32	4.16	14.47





Robustness





	StyleGAN2		StyleGAN-XL		StyleGAN3	
	C_θ	FID	C_θ	FID	C_θ	FID
Attacks	43.05	5.4	48.79	2.67	40.33	6.61
Black-box Attacks						
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
White-box Attacks						
Overwrite	4.78	8.34	4.91	8.83	4.73	9.71
RPT_{200}	4.91	5.47	4.52	3.52	4.59	6.65
RPT_{100}	4.44	5.56	4.21	3.90	4.47	6.75
RPT_{50}	4.38	8.07	4.38	15.32	4.16	14.47

Summary of Results

Watermarking can be robust against a Black-box attacker

No-box watermarking is not robust against a White-box attacker

PTW is scalable: (i) No training data, (ii) low compute, (iii) post-hoc application

Model-dependent key generation, optimize discoverability and adds obfuscation

PTW is applicable to other generator models (Latent Diffusion/Transformer-based)

Limitations

(IP Protection) We do not evaluate model extraction attacks

(From Scratch) We do not evaluate from-scratch trained image generators

(Non-cooperative Providers) Watermarking requires watermarking from all providers
(we did not evaluate collision attacks)

Check out the Paper for more Information

PTW: Pivotal Tuning Watermarking for Pre-Trained Image Generators

Nils Lukas, University of Waterloo
Florian Kerschbaum, University of Waterloo

arXiv:2304.07361v2 [cs.LG] 24 May 2023

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× 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

§To appear at USENIX Security 2023.

1

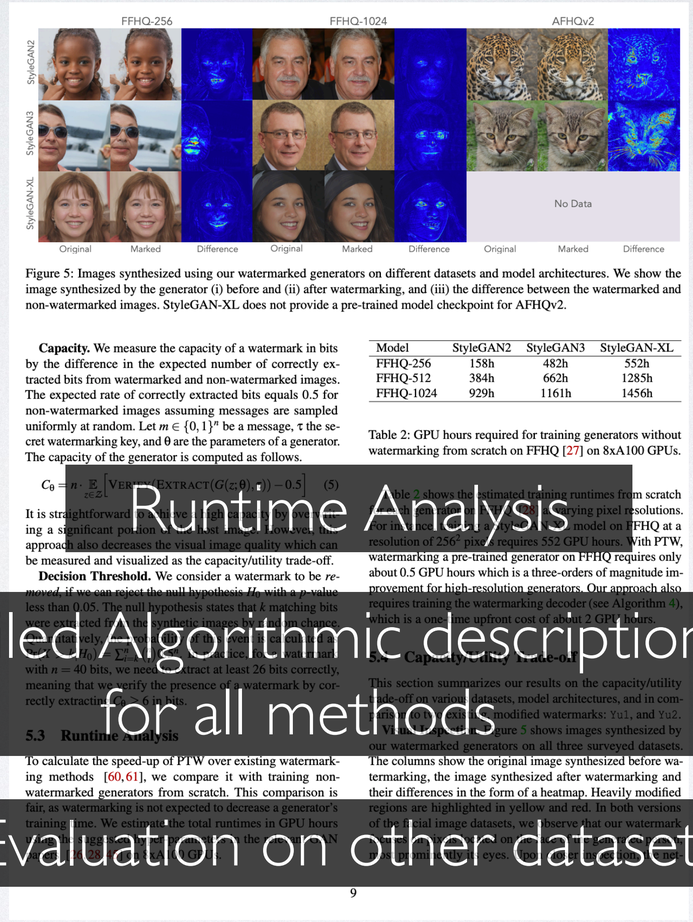


Figure 5: Images synthesized using our watermarked generators on different datasets and model architectures. We show the image synthesized by the generator (i) before and (ii) after watermarking, and (iii) the difference between the watermarked and non-watermarked images. StyleGAN-XL does not provide 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, τ the secret watermarking key, and θ are the parameters of a generator. The capacity of the generator is computed as follows.

$$C_\theta = n - \mathop{\mathbb{E}}_{z \in \mathbb{Z}^k}\left[\textsc{Verdict}(\textsc{Extract}(G(z;\theta), \tau)) - 0.5\right] \quad (5)$$

It is straightforward [illegible] ing 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. [illegible] $\sum_{i=0}^{k} \binom{n}{i} 0.5^n < 0.05$. For example, with $n = 40$ bits, we need to extract at least 26 bits correctly, meaning that we verify the presence of a watermark by correctly extracting [illegible] of 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 [illegible]

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 [illegible] resolutions. For the [illegible] model on FFHQ at a resolution of 256^2 pixels requires 552 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 with 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 [illegible] image datasets, we observe that our watermark [illegible] eyes. [illegible]

9

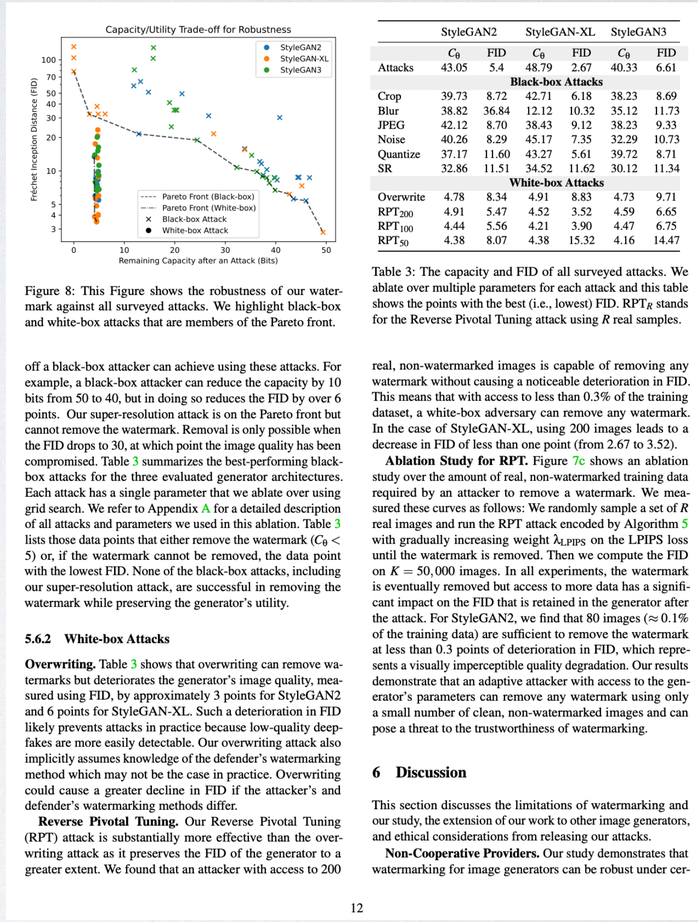


Figure 8: This Figure shows the robustness of our watermark against all surveyed attacks. We highlight black-box and white-box attacks that are members of the Pareto front.

	StyleGAN2		StyleGAN-XL		StyleGAN3	
	C_θ	FID	C_θ	FID	C_θ	FID
Attacks	43.05	5.4	48.79	2.67	40.33	6.61
Black-box Attacks						
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
White-box Attacks						
Overwrite	4.78	8.34	4.91	8.83	4.73	9.71
RPT_{200}	4.91	5.47	4.52	3.52	4.59	6.65
RPT_{100}	4.44	5.56	4.21	3.90	4.47	6.75
RPT_{50}	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_R stands for the Reverse Pivotal Tuning attack using R real samples.

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 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 λ_{LPIPS} on the LPIPS 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-

12

Runtime Analysis

Detailed Algorithmic descriptions for all methods

Evaluation on other datasets

Robustness with more attacks

PTW: Pivotal Tuning Watermarking for Pre-Trained Image Generators

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

Nils Lukas

Florian Kerschbaum

UNIVERSITY OF WATERLOO

CrySP
Cryptography, Security, and Privacy Research Group

GitHub - Source Code

Full Paper

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] Solaiman, Irene, et al. "Release strategies and the social impacts of language models." *arXiv preprint arXiv:1908.09203* (2019).

[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

Appendix

https://nilslukas.github.io