Frequency-aware GAN for Adversarial Manipulation Generation

Peifei Zhu, Genki Osada, Hirokatsu Kataoka, Tsubasa Takahashi
LINE Corporation
{peifei.zhu, genki.osada, jpz4219, tsubasa.takahashi}@linecorp.com

Abstract

Image manipulation techniques have drawn growing concerns as manipulated images might cause morality and security problems. Various methods have been proposed to detect manipulations and achieved promising performance. However, these methods might be vulnerable to adversarial attacks. In this work, we design an Adversarial Manipulation Generation (AMG) task to explore the vulnerability of image manipulation detectors. We first propose an optimal loss function and extend existing attacks to generate adversarial examples. We observe that existing spatial attacks cause large degradation in image quality and find the loss of high-frequency detailed components might be its major reason. Inspired by this observation, we propose a novel adversarial attack that incorporates both spatial and frequency features into the GAN architecture to generate adversarial examples. We further design an encoder-decoder architecture with skip connections of high-frequency components to preserve fine details. We evaluated our method on three image manipulation detectors (FCN, ManTra-Net and MVSS-Net) with three benchmark datasets (DEFACTO, CASIAv2 and COVER). Experiments show that our method generates adversarial examples significantly fast (0.01s per image), preserves better image quality (PSNR 30% higher than spatial attacks), and achieves a high attack success rate. We also observe that the examples generated by AMG can fool both classification and segmentation models, which indicates better transferability among different tasks.

1. Introduction

With the rapid development of advanced editing software, manipulated images are becoming more common on social media. Despite the positive aspects, there are possibilities that manipulated images are used to spread fake news and misleading information. Therefore, it is important to develop methods that can automatically detect manipulated images.

Various image manipulation detectors [5, 20, 32, 4] have been proposed and achieved high performance on manipulated image datasets [26, 7, 36, 16]. However, existing methods can be vulnerable to adversarial examples. For a manipulated image, adding imperceptible adversarial perturbations might cause the detector to detect a completely wrong result. Several works have already studied adversarial face forgery that fools the classification result of face forgery detector by the adversarial examples [15, 19, 14]. Since the detectors developed recently not only focus on classifying manipulated or authentic but also try to pinpoint the manipulated regions, we also go one step further to generate adversarial examples for the detectors that apply semantic segmentation. We name this task Adversarial Manipulation Generation (AMG). This topic has not been studied before and it is more difficult because instead of only fooling a class label, our target is to fool every pixel in both manipulated and authentic regions. We further show

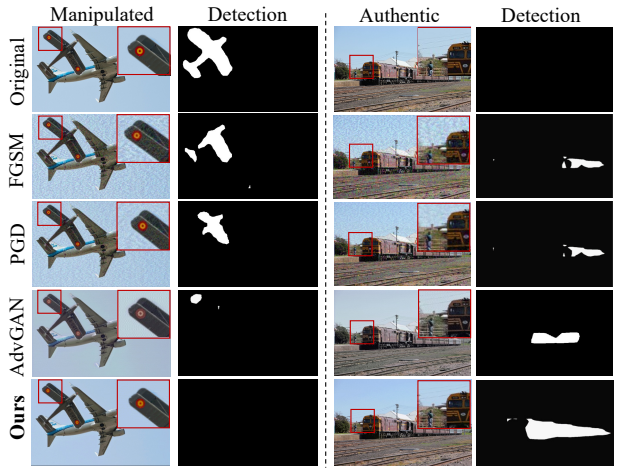


Figure 1. Adversarial examples generated under the proposed AMG task. For the manipulated image, the manipulations of the original image can be detected by a detector named MVSS-Net [5]. After applying attacks, the detector fails to correctly detect manipulations. For the authentic image, attacking the original image results in false-positive detection. Compared to previous attack methods, the examples generated by our method have much less noise and can successfully fool the detector at pixel-level.

the examples generated by AMG have better transferability among different tasks (details in Section 5.5).

As shown in Figure 1, images added with imperceptible perturbations generated by our task can successfully fool the detector. This task is also different from dense adversary generation [39, 3, 30] which is designed for segmentation and detection models under an untargeted attack setting. Our work considers applying targeted adversarial attacks on the manipulated and authentic images to explore the vulnerability of the image manipulation detectors. This is a specific scenario that the adversarial examples actually bring serious social problems. We first design a loss function that optimizes perturbation at pixel-level to generate such adversarial examples.

Algorithms such as FGSM [11] and PGD [25, 18] can be extended using our loss function to generate adversarial examples. However, these examples can be easily recognized by human eyes since the images are perturbated with visible noise. In addition, generating examples using iterative methods such as PGD can be time-consuming. To address these problems, we proposed a novel adversarial attack that generates perturbations using both frequency and spatial features to avoid image quality degradation. In addition, instead of optimizing each example against the target model, our method only needs to train the generator once, and then it can generate perturbation for any images extremely fast. The contributions of this work are three-fold:

- We explore the vulnerability of the image manipulation detectors by proposing an AMG task. We design a loss function and extend existing attacks to generate adversarial examples. Although these attacks can fool the detectors, the perturbations added to the image are visible to human eyes. We perform an analysis and find the loss of high-frequency components can be the major reason to cause the degradation of image quality.

- Inspired by the above observation, we propose an adversarial attack that incorporates both spatial and frequency features into the GAN architecture to generate adversarial examples. To preserve fine details, an encoder-decoder with skip connections of high-frequency components is also combined. Compared with previous attacks, our method generates more imperceptible perturbations for human observers.

- We evaluate our method on three manipulation detectors with three benchmark datasets, under both white-box and black-box settings. Experiments show that our method generates adversarial examples significantly fast, preserves better image quality, and achieves a high attack success rate. We further observe the examples generated by AMG have better transferability and can fool both classification and detection models.

2. Related Work

In this section, we briefly introduce the development of image manipulation generation and detection. Besides, we summarize recent adversarial attack methods that could be applied to image manipulation detectors.

2.1. Image Manipulation Generation

Image manipulation generation (also known as image forensic) has gained growing attention as it is getting easier to create fake images and might cause problems, such as spreading fake news and misleading information [34, 33]. Common image manipulation generation includes copy-move (copy a region and paste it in the same image), splicing (copy a region and paste it to another image) and inpainting (remove a region) [26, 22]. Several datasets [26, 7, 36] have been created and released, i.e. DEFACTO [26] is a recently released large-scale dataset, containing 149k images sampled from MS-COCO [21]. Recently, generative adversarial networks (GANs) [10, 45] have also become popular to generate forged images, especially in the field of face forgery generation [15, 19, 14]. Instead of only focusing on face manipulation, this work explores how adversarial attack affects manipulation generation that includes various types of images.

2.2. Image Manipulation Detection

Image manipulation detection can be considered as a classification task or a semantic segmentation task. In classification task, the goal is to distinguish the manipulated image from the real one. Methods such as [6, 1, 31] are proposed to learn the decision boundary between real and fake faces for face forgery detection. On the other hand, in semantic segmentation, the goal is to pinpoint manipulated regions at the pixel level. [20, 43] propose to implement a fully convolutional network (FCN) [23] to localize manipulations. ManTra-Net [37] designs a self-supervised learning task to learn robust image manipulation traces. MVSS-Net [5] exploits noise distribution and boundary artifacts surrounding manipulated regions to learn more generalizable features. Since this work mainly focuses on the semantic segmentation task, we selected one baseline method (FCN) and two state-of-the-art methods (ManTra-Net and MVSS-Net) for localizing manipulations and we demonstrate that our method can fool all of these detectors.

2.3. Adversarial Attack

Adversarial examples are inputs added with small perturbations to confuse a neural network. FGSM [11] is a well-known one-step attack method that uses gradients of the loss with respect to the input image. PGD [25, 18] is an iterative attack that performs FGSM with a smaller step size and clips the updated adversarial sample into a valid

range. Other methods such as CW [29], JSMA [28], MI-FGSM [8], are also widely used. However, such attacks using iterative optimization can be time-consuming, and each image should be optimized separately. Instead of optimizing each image, GAN-based methods [38, 17, 13] train a generator to learn the adversarial distribution by optimizing target loss and GAN loss. Once the generator is trained, it can generate perturbations for any input and is much faster than iterative-based methods. However, although the generated image can bypass the target model, it degrades the image quality and is visible to human eyes.

For adversarial examples in image manipulation detection, several works [9, 14, 19, 27, 44] were proposed to escape face forgery detection using adversarial attack. These methods mainly generate examples in the spatial domain. Recently, a few works [15, 24, 42] have been proposed to combine the frequency features to generate more imperceptible examples. However, these works are usually designed for the classification task, and the generation is optimization-based which can be time-consuming. We proposed a method that can generate high-quality adversarial images and is faster than most of the previous methods.

3. Adversarial Manipulation Generation

3.1. Problem Definition

Given an original image x and an image manipulation detector f which segments the manipulated and authentic region at pixel-level, we aim for generating adversarial examples that can completely fool the detector by adding imperceptible perturbations. We name this task Adversarial Manipulation Generation (AMG). The original image x can be either manipulated or authentic. In this work, we mainly focus on the targeted attack which contains two cases: 1) for the manipulated images, we add perturbations so that the detector fails to segment any pixels in the manipulated region, and 2) for the authentic images, we generate examples that the detector detects false-positive regions. Therefore, we design two types of target map: authentic map S_0 with all pixels labeled as authentic and manipulated map S_1 with the manipulation regions generated by watershed segmentation [2]. We select one random region from the watershed segmentation results and let it be the target manipulation region for each image. In order to determine which target map to use, we first input x into f to obtain a predicted map $\{f(x)\}$, then we apply Global Max Pooling (GMP) to obtain the image-level prediction result $GMP\{f(x)\}$ (authentic: 0, manipulated: 1).

Our loss function to generate the adversarial examples is as follows:

$$\mathcal{L}_{adv} = \begin{cases} \mathcal{L}_{DL}\left[f\left(x'\right), S_0\right] & GMP\{f(x)\} = 1 \\ \mathcal{L}_{DL}\left[f\left(x'\right), S_1\right] & GMP\{f(x)\} = 0 \end{cases} \quad (1)$$

where x' is the adversarial example generated by adding pixel-wise perturbation to the original image x. Since this is a semantic segmentation task, we use dice loss $\mathcal{L}_{DL}$ to calculate the difference between the predicted map and the target map. x' can be calculated by minimizing $\mathcal{L}_{adv}$, where x' should be as close as possible to x.

3.2. Extending Existing Attacks

Existing adversarial attacks can be extended to AMG using Equation 1. We selected three widely used attacks, FGSM, PGD, and AdvGAN, and extend them to generate the adversarial examples for AMG.

Fast Gradient Sign Method (FGSM) [11] is a one-step attack method that uses gradients of the loss with respect to the input image to calculate perturbations. The adversarial example generation can be written as:

$$x' = x - \varepsilon \cdot sign\left[\nabla_x \mathcal{L}_{adv}\left(x, S\right)\right] \quad (2)$$

Projected Gradient Descent (PGD) [25] is an iterative attack that performs FGSM with a smaller step size and projects the updated adversarial sample into a valid range, written as:

$$x'_{t+1} = Proj\left\{x'_t - \omega \cdot sign\left[\nabla_x \mathcal{L}_{adv}\left(x, S\right)\right]\right\} \quad (3)$$

AdvGAN [38] is a GAN-based attack that trains a generator to learn the adversarial distribution by maximizing the target loss and the GAN loss. A soft hinge loss is also incorporated to bound the magnitude of the perturbation. Therefore, the full objective can be expressed as $\mathcal{L}_{adv} + \alpha\mathcal{L}_{GAN} + \beta\mathcal{L}_{hinge}$. By solving the minimax game over the objective, the generator and discriminator can be optimized, and the adversarial example can be obtained by putting the original image into the generator.

Although these methods generate examples that can fool the manipulation detectors, the generated images can be easily recognized by human eyes since the images are perturbated with visible noise. Our work focuses on generating examples that can fool both target detectors and human eyes. Such attacks are more dangerous and might cause serious security problems.

3.3. Exploring Frequency Components

Previous works [40, 41, 35] have shown neural networks prefer to generate low-frequency signals that are more superficial in complexity. Therefore, part of the high-frequency signal might be lost during the feature extraction, and this results in generating images with noise and aliasing artifacts. In order to explore whether a similar phenomenon has occurred in generating the adversarial examples and thus design an optimized architecture, we first analyze the changes in frequency components when applying existing adversarial attacks.

The frequency components of the original image and the adversarial examples using existing attacks are visualized

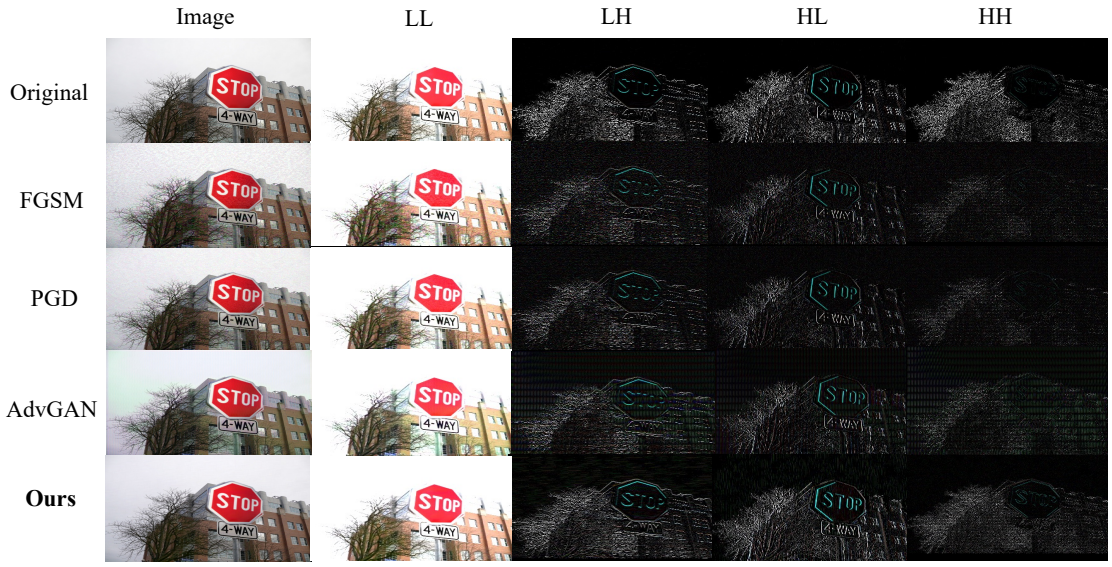


Figure 2. Visualization of the frequency components. Rows from top to bottom: original image, adversarial examples using FGSM, PGD and AdvGAN attack. Columns from left to right: image in the spatial domain, low-frequency component (LL), high-frequency horizontal component (LH), high-frequency vertical component (HL), high-frequency diagonal component (HH). For each component of the attack images, closer to that component of the original image indicates better preservation of the information.

in Figure 2 (row 1-4). We use wavelet transform to obtain the frequency components. The low-frequency component (LL) contains the general structure and most of the color data, while the high-frequency component (LH, HL, HH) contains rich details like the edges of the contents and image texture. By comparing the components between the original image and adversarial examples, we find that many details of the high-frequency components such as the branches of the tree, and the edge of the building windows, are lost, meanwhile, artifacts appear in the background. We believe the loss of details and the increasing artifacts in the high-frequency components are major reasons that make adversarial perturbations visible to human eyes.

4. Frequency-aware GAN Attack

4.1. Architecture Overview

Inspired by the above observation, we propose an attack method to generate high-quality adversarial examples for AMG. As shown in Figure 3, the architecture of our method has three components: a generator G, a discriminator D, and a target model f. We focus on designing a G that can generate imperceptible perturbation. G consists of an encoder and a decoder to generate pixel-level perturbations. The encoder extracts high-level feature representation from the input image, and the decoder uses the representation to generate perturbations. In this work, instead of directly perturbating the image in the spatial domain, we transform the input image into the frequency domain, combine the frequency feature and the spatial feature and add noise on the mixed domain to avoid degradation of image quality. The perturbations are then added to the input image to obtain an adversarial image.

The discriminator D is a classifier that distinguishes the original input image from the adversarial image. A minimax GAN loss $\mathcal{L}_{GAN}$ can be obtained, where D tries to distinguish the adversarial image and the input image, and G tries to generate perturbations that are indistinguishable by the D. To obtain the adversarial example, we first apply the white-box attack, where the target model f is known and used to train the model. The input of f is the adversarial image, and the output is the predicted segmentation. An adversarial attack loss $\mathcal{L}_{adv}$ can be calculated between the predicted and target result.

4.2. Frequency-aware Generator

We propose a generator that combines both spatial and frequency features as well as incorporates skip connections of the high-frequency components to generate imperceptible adversarial examples. The architecture of the generator is shown in Figure 3.

Encoder. The encoder consists of convolutional layers (conv), pooling layers, and discrete wavelet transform layers (dwt). The first conv layer extracts spatial features from the input image and passes the feature to the dwt layer. We used 2D discrete wavelet transform to decompose the features into the low-frequency component (L_i) and the high-frequency component (H_i). These components are concate-

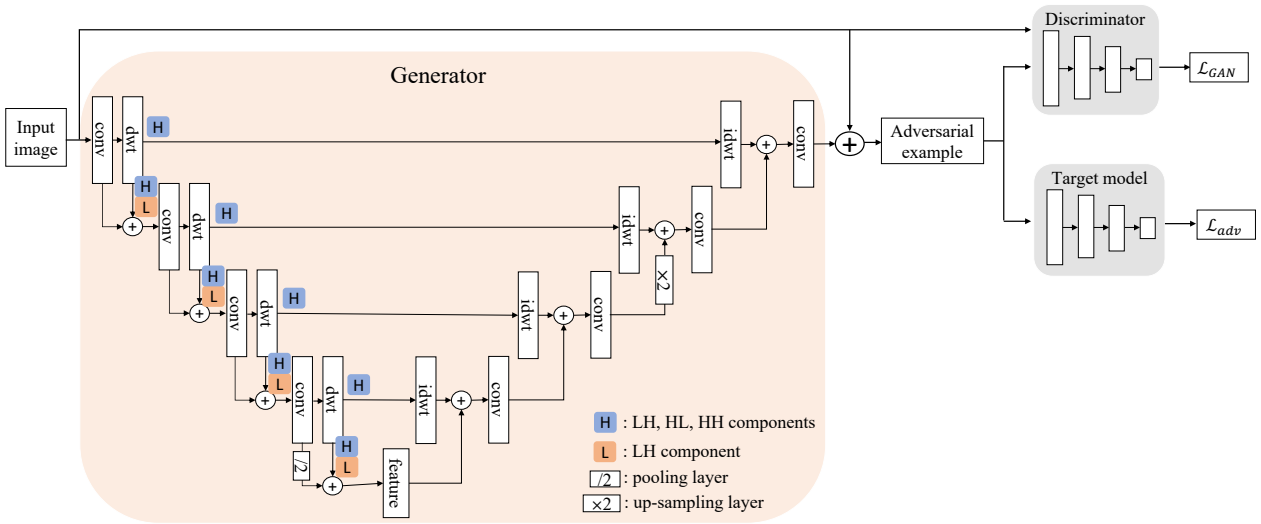


Figure 3. Overview of our frequency-aware GAN attack.

nated channel-wise, denoted as $concat(L_i, H_i)$. On the other hand, the spatial feature extracted by conv layer is downsampled and then combined with the frequency feature $concat(L_i, H_i)$. Finally, the features containing both spatial and frequency information are sent to the next conv layer. We construct 4-level layers to extract the high-level feature representation.

Decoder. The decoder consists of conv layers, upsampling layers, and inverse discrete wavelet transform layers (idwt). To avoid losing fine details, we conduct skip connections of the high-frequency components from the encoder to the decoder. This helps the decoder generate perturbations with more details and less noise, and thus makes the generated image less distinguishable by human eyes. The processing of the decoder is as follows. The feature generated by the encoder is upsampled to obtain feature A. On the other hand, 2D inverse discrete wavelet transform is performed on the high-frequency feature directly from the encoder, then we obtain feature B. These two features are combined as $A + B$ and sent to the conv layer. We also construct 4-level layers to decode the feature and generate the perturbation for the input image.

4.3. Optimization

As in the white-box attack setting, the target model f is known and fixed, we need to optimize the generator G and the discriminator D during training. The total loss consists of two terms: adversarial attack loss $\mathcal{L}_{adv}$ and GAN loss $\mathcal{L}_{GAN}$. $\mathcal{L}_{adv}$ can be obtained by summarizing the loss in Equation 1 for a set of input image and $\mathcal{L}_{GAN}$ can be written as:

$$\mathcal{L}_{GAN} = \mathbb{E}_x \log D\left(x\right) + \mathbb{E}_x \log\left\{1 - D\left[x + clip(G\left(x\right))\right]\right\}, \tag{4}$$

where $\mathbb{E}_x = \mathbb{E}_{x \sim \mathcal{P}_{data}}$ and $\mathcal{P}_{data}$ is original data distribution, and $clip$ stands for clipping $G(x)$ into a valid range. The adversarial example can be obtained by $x + clip(G\left(x\right))$.

Finally, the objective function for training our model can be written as:

$$\min_G \max_D V(D, G) = \mathcal{L}_{adv} + \gamma \mathcal{L}_{GAN}, \tag{5}$$

where γ is a weight to control the importance of two losses. By minimax game between the G and D, optimal parameters of the model can be obtained.

5. Experiments

In this section, we evaluate our proposed methods under both white-box and black-box attack settings. In the white-box setting, we first train the generator and discriminator using a target model, then during the attack, we put images into the trained generator to generate adversarial examples for that target model. In the black-box setting, we use a transferability-based attack, where one target model is selected to train the generator and obtain adversarial examples, then these adversarial examples are directly used to attack another unknown model. We perform evaluations on several image manipulation datasets and compare our method with existing adversarial attack methods, FGSM, PGD and advGAN. We run all experiments on an NVIDIA A100 80GB GPU.

5.1. Experimental Settings

Datasets. We evaluate the proposed method on three image manipulation datasets, DEFACTO [26], CASIAv2 [7] and COVERAGE [36]. DEFACTO contains 149k images with copy-move, splicing, and inpainting manipulations. This dataset is constructed over MS-COCO [21] and

contains natural images with various contents, such as natural scenes, objects, etc. As we need both manipulated and authentic images, similar to [5], we randomly sample 64k manipulated images from DEFACTO and 20k authentic images from MSCOCO to construct a dataset. CASIAv2 is a natural image forgery dataset that contains 4795 images, 1701 authentic and 3274 forged with copy-move and splicing manipulations. COVERAGE (COVER) contains 100 copy-move forged images and their originals with similar but genuine objects. The ground-truth masks are provided in all datasets. Since AdvGAN and our method need training data, we further split each dataset into half and half. We use the half dataset to train AdvGAN and our method and use the other half to evaluate all attack methods.

Target models. For the image manipulation detectors, we selected three models, FCN [23], ManTra-Net [37] and MVSS-Net [5]. These models are all trained on part of the DEFACTO dataset, and their performance has been compared in previous works [5]. As all these detectors with pretrained models and optimized parameters are already available, we directly use them as our target models.

Evaluation metrics. We compute pixel-level F1 (pF1) for semantic segmentation of the manipulated region. Since we also have authentic images, we report image-level F1 (imF1) as well. Please note authentic images are only used to calculate im-F1, while manipulated images are used to calculate both metrics. As the purpose of this work is to explore the robustness of the manipulation detectors by the adversarial attack, we also use attack success rate (ASR) as one of the evaluation metrics. We define "success" as the opposite of the ground-truth which means 1) for manipulated images, every pixel is detected as authentic, and 2) for authentic images, part of the image is detected as manipulated. The proportion of the image that is successfully attacked over the total test dataset is denoted as the ASR.

Implementation details. The architecture of the generator is shown in Figure 3. For the discriminator, we use a similar architecture to the discriminator in AdvGAN [38]. We set the weight of GAN loss $\gamma = 0.1$ and clipping range $clip = 0.1$. During the model training, we set the batch size to 128 and the learning rate to 0.001. The input image size of three target models is $512 \times 512 \times 3$. For the FGSM and PGD attack, we set maximum perturbation $\epsilon = 0.05$.

5.2. White-box Attack

In this setting, we generate adversarial examples against each target model. We evaluate our method using different manipulation detectors with different datasets. We first measure the performance of original images using three manipulation detectors and then apply four types of adversarial attacks to see the changes in performance. The results of pF1, imF1, and ASR are shown in Table 1. We observe that for three different image manipulation detectors, p-F1 and im-F1 largely drop when applying adversarial attacks. A similar trend can be observed for all datasets. These results show that even though the selected detectors perform well on specific manipulation datasets, they can still be vulnerable to adversarial examples that only have small changes from the original ones.

On the other hand, compared with the existing attacks, the ASR of our method is higher than FGSM and slightly higher than PGD and AdvGAN for most of the cases. We also decompose the adversarial examples generated by our method into frequency components, and an example is shown in Figure 2 (row 5). Compared with existing attack methods, our method can preserve most of the details, especially the high-frequency components such as the branches of the tree, and the edge of the building windows, as well as keep noises and artifacts as fewer as possible. The running time of generating adversarial images for each dataset is shown in Table 1. For generating one image, the average running time is: FGSM 0.09s, PGD 20s, AdvGAN 0.02s, ours 0.01s. Our method is significantly fast than iterative-based attack methods.

5.3. Black-box Attack

We also conduct experiments to evaluate the transferability of our attack method. We use one target model to train the generator and obtain adversarial examples, then directly use these examples to attack other target models.

We use the DEFACTO dataset, where half of the data is used to train AdvGAN and our method, and the other half is used to generate adversarial examples. Then these generated examples are sent to the target models to obtain the ASR. The results are shown in Table 2. The model used to generate the adversarial example is denoted as the trained model (Train), and the model used to calculate ASR is denoted as the target model (Target). For better comparison, we also add the result of the white-box setting (the trained model and the target model are the same). We denote FCN, ManTra-Net, and MVSS-Net as model A, B, and C.

From Table 2, we observe that for FGSM and PGD attacks, the ASR of the black-box attack drops more than 10% compared to the white-box attack in several cases, while AdvGAN and our method seem to have better transferability. This is probably because FGSM and PGD are optimized-based attacks where each image should be optimized separately against the target model, while AdvGAN and our method try to learn and approximate the distribution of the datasets. Another interesting finding is that for different target models, the adversarial examples generated by models with more complex architecture (i.e. MVSS-Net = model C) seem to have better transferability than the examples generated by the baseline model (FCN = model A). We consider the reason as complex models usually use the baseline model as their backbone, therefore the examples

Data	Method	FCN			ManTra-Net			MVSS-Net			Run Time
		pF1	*imF1*	*ASR*	*pF1*	*imF1*	ASR	*pF1*	*imF1*	*ASR*	
DEFACTO	Origin	0.68	0.65	-	0.70	0.66	-	0.72	0.68	-	-
	FGSM	0.28	0.25	50.5%	0.27	0.29	48.2%	0.28	0.27	50.2%	2.5h
	PGD	0.21	0.19	58.5%	0.19	0.18	58.9%	0.23	0.19	57.2%	470h
	AdvGAN	**0.17**	0.16	60.2%	0.18	0.17	59.9%	**0.20**	**0.18**	**60.8%**	30min
	Ours	**0.17**	**0.15**	**60.5%**	**0.16**	**0.16**	**60.6%**	0.21	**0.18**	60.5%	**15min**
CASIAv2	Origin	0.59	0.50	-	0.60	0.58	-	0.69	0.62	-	-
	FGSM	0.29	0.26	48.2%	0.26	0.25	48.5%	0.30	0.30	44.2%	10min
	PGD	0.21	0.18	56.5%	0.20	0.21	50.7%	0.23	0.20	51.4%	36h
	AdvGAN	**0.18**	**0.16**	**59.7%**	0.19	0.19	55.8%	0.20	0.15	58.8%	2.1min
	Ours	0.20	0.17	58.8%	**0.18**	**0.16**	**58.2%**	**0.18**	**0.14**	**60.2%**	**1.1min**
COVER	Origin	0.48	0.39	-	0.44	0.41	-	0.63	0.55	-	-
	FGSM	0.20	0.18	50.0%	0.19	0.17	52.3%	0.22	0.24	50.1%	18min
	PGD	**0.14**	0.14	61.1%	0.14	0.14	55.3%	0.16	0.15	60.3%	68min
	AdvGAN	**0.14**	0.13	61.5%	0.14	0.12	57.1%	0.15	0.15	60.6%	4.1s
	Ours	**0.14**	**0.12**	**61.9%**	**0.13**	**0.11**	**59.1%**	**0.13**	**0.12**	**62.2%**	**2.5s**

Table 1. The performance of different attacks under AMG task. The evaluation is performed on three datasets (DEFACTO, CASIAv2 and COVER) with three manipulation detectors (FCN, ManTra-Net and MVSS-Net) as the target models. For the attack method, higher ASR (lower $pF1$ and $imF1$) means better performance. Run time shows the generation time for all examples in that dataset.

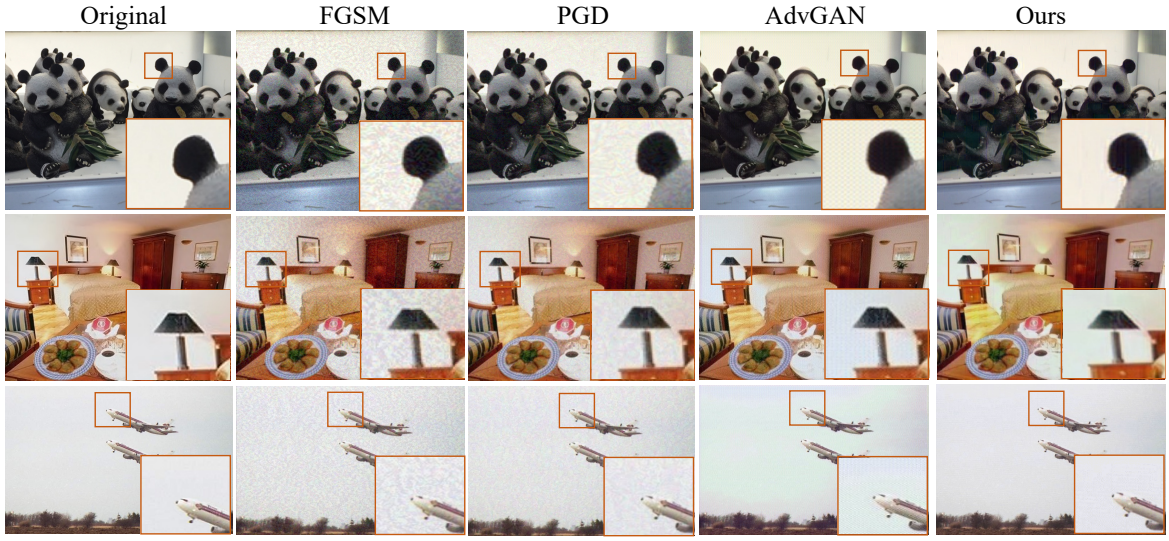


Figure 4. Comparison of generated adversarial examples under different attack methods.

generated by the complex model might naturally have the ability to attack baseline models. These findings give us a new perspective to generate adversarial examples with better transferability.

5.4. Image Quality Assessment

In this section, we show that our method could keep high image quality while fooling the target model. We evaluate the generated examples qualitatively and quantitatively on MVSS-Net with the CASIAv2 and COVER datasets. For qualitative evaluation, the visualization examples of four attack methods are shown in Figure 4. We can observe that compared with the existing attacks, the adversarial examples generated by our method have much less noise, look more natural and are more imperceptible to human eyes. For quantitative evaluation, we use metrics MSE, PSNR, and SSIM to calculate the difference between the original image and the adversarial example. As shown in Table 3, the image quality of the adversarial examples generated by our method outperforms others by a large margin. This result suggests that our method is able to keep high image quality while successfully attacking target models.

Train	Target	FGSM	PGD	AdvGAN	ours
A	A	50.5%	58.5%	60.2%	**60.5%**
	B	39.3%	43.3%	50.2%	**55.1%**
	C	42.6%	40.2%	52.1%	**58.3%**
B	B	48.2%	58.9%	59.9%	**60.6%**
	A	43.6%	50.7%	**55.1%**	53.1%
	C	37.3%	46.2%	50.6%	**52.4%**
C	C	50.2%	57.2%	**60.8%**	60.5%
	A	45.3%	52.3%	54.2%	**58.4%**
	B	42.3%	52.1%	56.3%	**62.2%**

Table 2. Attack success rate under the black-box attack settings. First column (train) is the model we use to generate adversarial example, and second column (target) is the target model to attack. A, B and C represent FCN, ManTra-Net and MVSS-Net model.

Metric	FGSM	PGD	AdvGAN	ours
MSE (↓)	0.71	0.69	0.59	**0.50**
PSNR (↑)	26.4	30.4	28.9	**40.1**
SSIM (↑)	0.61	0.79	0.71	**0.98**

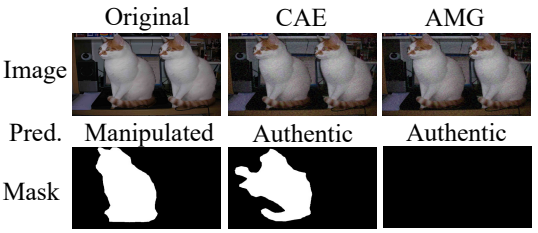
Table 3. Image quality assessment of the adversarial examples generated by different attacks. Our method has significantly better image quality than other attacks.

5.5. Analysis

In this section, we first analyze the advantage of the proposed AMG task and then design four ablations to study the mechanism of the proposed attack.

To analyze the advantage of the proposed AMG task, we compared the adversarial examples generated under AMG and traditional classification task (CAE), shown in Figure 5. PGD attack is used to generate examples for both CAE and AMG task. We randomly selected 1000 manipulated images and 1000 authentic images from DEFACTO, CASIAv2 and COVERAGE datasets for evaluation. Two target models, a classification model named ResNet-50 [12] and a segmentation model named FCN [20] are selected as the target models. The adversarial examples generated under CAE are used to attack both ResNet-50 (white-box attack) and FCN (transferability-based attack), and vice versa. We observe that although the examples generated under CAE have a high ASR for attacking the classification model, the ASR drops largely when they are used to attack the detection model. On the other hand, the examples generated under our AMG task is able to attack both models, which indicates better transferability among different tasks.

To analyze the behavior of each component in the proposed method, We evaluated four ablations : 1) we use discrete wavelet transform (DWT) but remove the skip connection (SC) when generating the perturbations, 2) we only apply the SC of the low-frequency components, 3) we remove both DWT and SC, 4) our method. The results are shown in Table 4. By comparing 1st and 3rd rows, we observe that using the frequency features improves PSNR and SSIM by a large margin. By comparing the types of SC (1st, 3rd and 4th rows), we also observe that preserving high-frequency components have the most effect for improving the image quality. These quantitative results show the effect of two main components of the proposed method.



ASR-class	-	98.0%	85.4%
ASR-pixel	-	8.9%	57.6%

Figure 5. Comparing the transferability of the adversarial example generated by AMG and classification (CAE) task. Prediction (Pred.) and mask are obtained from ResNet-50 and FCN, respectively. ASR-class and ASR-pixel stand for the attack success rate of classification accuracy and pixel-level detection accuracy.

DWT	SC	ASR (↑)	PSNR (↑)	SSIM (↑)
✓	×	59.8%	33.5	0.84
✓	Low	60.0%	35.5	0.89
×	×	58.6%	30.8	0.74
✓	High	60.5%	40.0	0.97

Table 4. Ablation study of the proposed method on DEFACTO dataset with MVSS-Net detector. DWT is discrete wavelet transform, and SC is skip-connection (low or high means low-/high-frequency component).

6. Conclusion

In this work, we propose an AMG task to explore the vulnerability of image manipulation detectors. We extend the existing attacks and explore the changes in frequency components during adversarial attacks. Moreover, we propose a novel adversarial attack that incorporates both spatial and frequency features into the GAN architecture to generate imperceptible examples. Experiments show that our method generates adversarial examples significantly fast and preserves better image quality while achieving a high attack success rate. This work shows the vulnerability of current image manipulation detectors and suggests more robust detectors are needed to correctly detect the manipulations. Adversarial defense that can protect the detectors from various attacks is an interesting topic in future work.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018.

[2] Jesús Angulo and Dominique Jeulin. Stochastic watershed segmentation. In *ISMM (1)*, pages 265–276, 2007.

[3] Victor Besnier, Andrei Bursuc, David Picard, and Alexandre Briot. Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15701–15710, 2021.

[4] Ivan Castillo Camacho and Kai Wang. Convolutional neural network initialization approaches for image manipulation detection. *Digital Signal Processing*, 122:103376, 2022.

[5] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14185–14193, 2021.

[6] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 5781–5790, 2020.

[7] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. In *2013 IEEE China Summit and International Conference on Signal and Information Processing*, pages 422–426. IEEE, 2013.

[8] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[9] Apurva Gandhi and Shomik Jain. Adversarial perturbations fool deepfake detectors. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Weiwei Hu and Ying Tan. Generating adversarial malware examples for black-box attacks based on gan. In *Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II*, pages 409–423. Springer, 2023.

[14] Shehzeen Hussain, Paarth Neekhara, Malhar Jere, Farinaz Koushanfar, and Julian McAuley. Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3348–3357, 2021.

[15] Shuai Jia, Chao Ma, Taiping Yao, Bangjie Yin, Shouhong Ding, and Xiaokang Yang. Exploring frequency adversarial attacks for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4103–4112, 2022.

[16] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2889–2898, 2020.

[17] Guoqing Jin, Shiwei Shen, Dongming Zhang, Feng Dai, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3842–3846. IEEE, 2019.

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[19] Dongze Li, Wei Wang, Hongxing Fan, and Jing Dong. Exploring adversarial fake images on face manifold. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5789–5798, 2021.

[20] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8301–8310, 2019.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[22] Kai Liu, Junke Li, and Syed Sabahat Hussain Bukhari. Overview of image inpainting and forensic technology. *Security and Communication Networks*, 2022, 2022.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] Cheng Luo, Qinliang Lin, Weicheng Xie, Bizhu Wu, Jinheng Xie, and Linlin Shen. Frequency-driven imperceptible adversarial attack on semantic similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15315–15324, 2022.

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[26] Gaël Mahfoudi, Badr Tajini, Florent Retraint, Frederic Morain-Nicolier, Jean Luc Dugelay, and PIC Marc. Defacto: Image and face manipulation dataset. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[27] Paarth Neekhara, Brian Dolhansky, Joanna Bitton, and Cristian Canton Ferrer. Adversarial threats to deepfake detection: A practical perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 923–932, 2021.

[28] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[29] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[30] Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li. Semanticadv: Generating adversarial examples via attribute-conditioned image editing. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 19–37. Springer, 2020.

[31] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

[32] Zenan Shi, Chaoqun Chang, Haipeng Chen, Xiaoyu Du, and Hanwang Zhang. Pr-net: progressively-refined neural network for image manipulation localization. *International Journal of Intelligent Systems*, 37(5):3166–3188, 2022.

[33] Rahul Thakur and Rajesh Rohilla. Recent advances in digital image manipulation detection techniques: A brief review. *Forensic science international*, 312:110311, 2020.

[34] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020.

[35] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8684–8694, 2020.

[36] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage—a novel database for copy-move forgery detection. In *2016 IEEE international conference on image processing (ICIP)*, pages 161–165. IEEE, 2016.

[37] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2019.

[38] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

[39] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017.

[40] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[41] Mengping Yang, Zhe Wang, Ziqiu Chi, and Wenyi Feng. Wavegan: Frequency-aware gan for high-fidelity few-shot image generation. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022.

[42] Jiutao Yue, Haofeng Li, Pengxu Wei, Guanbin Li, and Liang Lin. Robust real-world image super-resolution against adversarial attacks. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5148–5157, 2021.

[43] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1053–1061, 2018.

[44] Tianfei Zhou, Wenguan Wang, Zhiyuan Liang, and Jianbing Shen. Face forensics in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5778–5788, 2021.

[45] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pages 597–613. Springer, 2016.