Information Sciences 683 (2024) 121267

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Evasion on general GAN-generated image detection by disentangled representation

Patrick P.K. Chan [a,*], Chuanxin Zhang [b], Haitao Chen [c], Jingwen Deng [c], Xiao Meng [d], Daniel S. Yeung [e]

[a] Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou, 511442, China
[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China
[c] School of Future Technology, South China University of Technology, Guangzhou, 511442, China
[d] Huya Inc, Guangzhou, 510006, China
[e] Hong Kong, China

ARTICLE INFO

Keywords:
GAN-generated image detection
Evasion attack
Disentangled representation

ABSTRACT

Images generated by the Generative Adversarial Network (GAN) are too realistic to be distinguished by humans. Recently, some detection methods have been proposed to distinguish between generated and real images. However, these methods rely on specific detection techniques and can be easily detected by other types of detection methods. This study aims to investigates the security of the GAN-generated image detection method by devising a method to evade general detection. The features related and unrelated to differentiating between real and generated images are disentangled by a GAN model in our model. The unrelated features contain information about the image content, while the related feature provides useful information for identifying generated images. Our method then camouflages a generated image by using its unrelated features and the related features of real images. The main advantages of our model include its ability to generalize to different detectors and adapt to the prior information about detectors. Experimental results confirm the superior evasion capability of our proposed method compared to other detector-dependent and independent methods across different popular detection methods.

1. Introduction

Generative Adversarial Networks (GANs) [1] have achieved excellent results in many applications of image generation, for example, generating non-existing images [2], changing image attributes [3], and styles [4]. Some studies [5] indicate that a generated image is too realistic to be distinguished by humans. This may raise a security issue because realistic images can be used to make fake news or generate forged criminal evidence.

To verify the authenticity of images, detection methods for the generated images are proposed. Some methods attempt to detect artifacts, *e.g.,* the pixel artifacts [6], and frequency domain artifacts [7–9] of an image. The generation method and image specification (*e.g.,* resize and compress) dependence is a disadvantage, *i.e.,* the performance of these methods using different image generation methods or image specifications drops dramatically [10]. Another type of detection determines whether an image is natural according to its high-level features, for example, human behavior, appearance [11], head poses [12], physiological signals [13] and eye blinking

* Corresponding author.
E-mail addresses: patrickchan@ieee.org (P.P.K. Chan), im.chuanxin@hotmail.com (C. Zhang).

https://doi.org/10.1016/j.ins.2024.121267
Received 1 August 2023; Received in revised form 17 June 2024; Accepted 24 July 2024

Available online 5 August 2024

[14]. Although high-level features are less sensitive to the image specifications, temporal information is usually required. As a result, these detection methods are primarily utilized to identify fake videos and focus on specific content [15].

In the arms race between generators and detectors, some methods increase the concealment of generated images by applying a detection method [16,17]. However, these methods may only successfully evade the particular detection but not others. In addition, the assumption of knowing detection methods may not be realistic in some applications. For example, Twitter removes synthetic and manipulated media that purposely mislead people, but its identification mechanism is not public [18].

This study aims to investigate the security issues of existing GAN-generated image detection methods by devising a model to enhance the concealment of a generated image. The image generated by our model can evade general detection by substituting fake attributes with those extracted from real images. During this process, our model operates independently of any information from the detector. Both features related and unrelated to the distinction between real and generated images are disentangled by the extractors. The unrelated features may contain the structure and semantic information related to the content of an image, while related features provide discriminative information for generated image identification. A GAN-generated image is camouflaged by using its unrelated features and the real-related features extracted from real images. We expect the image after processing by our model to be similar to the one before processing and also to have the characteristics of real images. The revised image is not manipulated according to any detection methods, but only relies on real and fake images we camouflage. Therefore, over-training for particular detections is avoided and the evasion ability should be generalized. Moreover, since our method does not focus on particular detection methods, no prior information about detectors is required. In a scenario in which prior information is obtained, our model can adapt the information to increase evasion ability. Our proposed method is evaluated and compared to the State-Of-The-Art (SOTA) methods experimentally. The results confirm that our method outperforms other methods in terms of evasion and image quality. This study proposes a disentangled representation framework for evasion attacks, which may reduce detection performance.

The contributions of this study include

- An evasion method for GAN-generated image detection is proposed in the framework of disentangled representation. The concealment of generated image increases as its features related to fake images are replaced with the features of real images.
- Since the proposed evasion method is independent of a detection method, no prior information is required, and the evasion ability of our camouflaged images can be generalized to any detection.
- The flexibility of the GAN model allows our method to adapt prior detector information. Evasion ability can be enhanced by knowing the type of detectors used.

The rest of this manuscript is organized as follows. Section 2 describes the ideas related to this study, including GAN-based image generation methods, generated image detection methods, and existing countermeasures. Our proposed model is introduced in Section 3. The experimental results are given and discussed in Section 4, and the conclusions and future works are presented in Section 5.

2. Related works

2.1. GAN-based image generation

A generative problem is formulated as an adversarial game between discriminator and generator in a GAN model. The discriminator distinguishes real data from fake one, while the generator synthesizes realistic data from the input, with the aim to mislead the discriminator. The minimax objective endows the generator's effective ability. Progressive growth GAN (ProGAN) [19] synthesizes high-resolution images by incremental enhancing the discriminator and generator networks during training. The training begins with low-resolution images and increases the resolution gradually. SNGAN [20] proposes spectral normalization to stabilize the training of the discriminator by limiting the gradient of the model to a range. CramerGAN [21] employs the probability metric of the Cramer distance instead of the Wasserstein distance, which exhibits superior properties of sum invariance, scale sensitivity, and unbiased sample gradients.

The gradient estimator used in the optimization process of MMDGAN [22] is unbiased and faster than Wasserstein-GAN. StyleGAN2 [2] solves the problem in which the generated images are prone to water drop artifacts, while StyleGAN3 [23] solves the absolute pixel coordinate dependence problem of StyleGAN2 by making images fully equivalent to translation and rotation even at sub-pixel scales and greatly improves the quality of image synthesis.

2.2. GAN-generated image detection

The detection methods can be classified into two types according to the feature level [24]. High-level feature-based methods determine whether an image is real according to semantic information, for example, heartbeat [13], head position [12], and habitual actions [15], while the low-level methods are based on artifact created by a generation process. A few low-level feature-based methods focus on the pixel domain [25,6], and most of them [7,26] extract the frequency features since artifacts can be more easily identified in the frequency domain. Several explicit indicators, *e.g.,* co-occurrence matrix [27], and color saturation [28], are proposed to capture the difference between generated and real images. Common patterns generated by filtering and nonlinear processes in a GAN method, called GAN fingerprint [10], are considered to be leveraged in the detection.

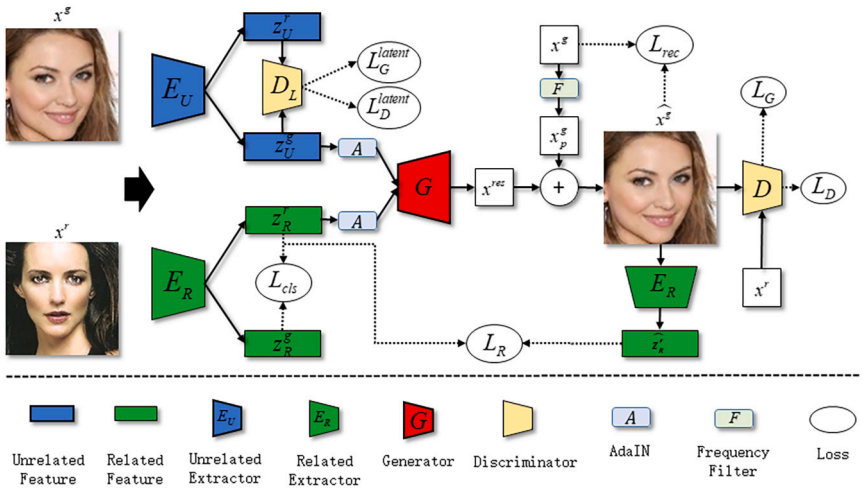

Fig. 1. The real image and generated image will go through E_R and E_U, respectively. Specifically, x^r is encoded to z^r_U and z^r_R, while x^g is encoded to z^g_U and z^g_R. z^r_U and z^g_U are trained not to contain label information, but z^r_R and z^g_R are trained to contain label information. By combining z^r_R and z^g_U, this study generates a residual image that contains the related features of real images but not those of the generated images, then get the camouflaged image $\hat{x}^g$ by adding the residual image to the filtered original generated image x^g_p.

2.3. Evasion of GAN-generated image detection

Evasion methods of adversarial learning [29,30] can be applied to mislead generated image detection methods. An imperceptible adversarial perturbation crafted according to a targeted model to mislead its decision is added to a generated image [31]. One of most popular attack strategies modifies a sample according to the gradient of the decision function [32]. In scenarios of partial knowledge, where complete information about a classifier is not accessible, images are usually camouflaged based on a surrogate model [33], multiple models [34] or a series of queries [35]. However, all these attack methods some knowledge of targeted model, which may not be realistic in some applications. The data representation of real images is extracted by *PCA* [36] and *KSVD* [37] in which a generated image is remarkably different from the real images. Recent studies [38,39] attempt to eliminate a range of traces, for example, spatial anomalies, spectral differences, and noise fingerprints, to evade detection. One advantage of these methods is their detector independence, which allows generalization to any detection methods. [40].

3. A proposed method to attack GAN-generated image detectors

The model aims to manipulate a GAN-generated image by replacing its fake characteristics with real ones. Our method first trains two encoders to disentangle the features related and unrelated to generated image identification. The generated image is then manipulated using its unrelated features, containing its content information, and real-related features of real images. The manipulated image should be able to evade general generated image detection.

3.1. Model component

Given a dataset $D = \{\mathbf{x}^y_1, \mathbf{x}^y_2, ..., \mathbf{x}^y_n\}$. $\mathbf{x}^y_i$ denotes the i^{th} image. $y \in \{\mathrm{r}, \mathrm{g}\}$, where r and g indicates whether $\mathbf{x}^y_i$ is a real and generated image, respectively. The goal of our study is to train a model f mapping a generated sample $\mathbf{x}^{\mathrm{g}}$ to $\hat{\mathbf{x}}^{\mathrm{g}}$, where $d(\hat{\mathbf{x}}^{\mathrm{g}}) = \mathrm{r}$ and d is a generated image detector. As a result, a sample manipulated by our model can evade detection.

Fig. 1 illustrates three kinds of components, including two encoders, one generator, and two discriminators, in training. Their detailed structures are shown in Fig. 2. The features related and unrelated to generated image identification are disentangled by two encoders called related (E_R) and unrelated (E_U) extractors. We expect that the semantic and structural content information of an image will be extracted by E_U. The generator G creates a residual image. The discriminator D determines whether an image is a generated image, and D_L verifies E_U does not extract related information. The details of each component are discussed in the following subsections.

3.1.1. Encoders

An image is extracted by two encoders into two parts, *i.e.,* the related (z_R) and unrelated (z_U) features. We use z^r_R (z^g_R) to denote the latent features of a real image (a generated image) of E_R and z^r_U (z^g_U) of E_U. To minimize the discriminant information in z_U, adversarial learning is employed. D_L identifies the generated images according to the relevant information provided by z_U. On the other hand, E_U is designed to reduce the accuracy of D_L in order to diminish the discriminative power of z_U. As a result, it is expected that the related information contained in z_U will be suppressed. L^{latent}_D and L^{latent}_G are minimized in the training of D_L and E_U respectively:

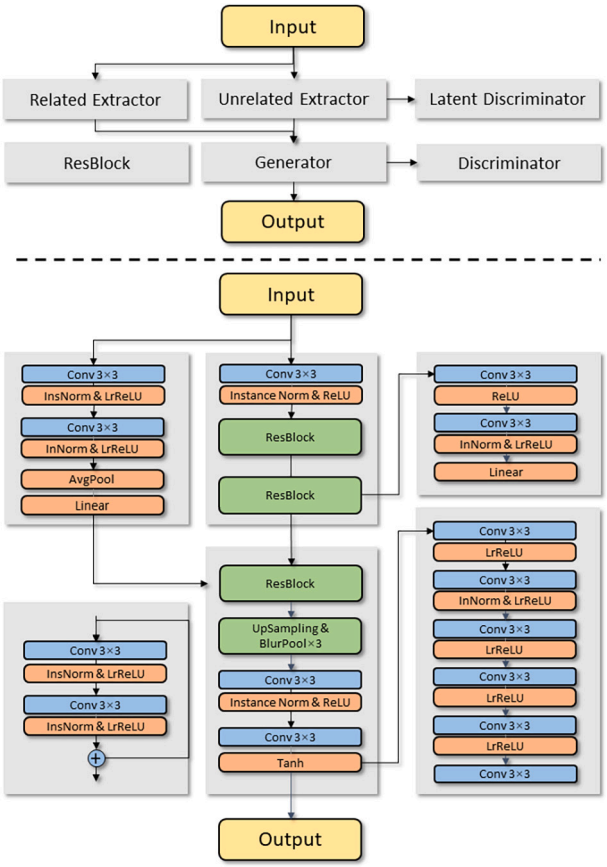

Fig. 2. The upper part of the figure shows the overall structure of the model, and the detailed structure corresponding to each network is shown in the lower part of the picture.

$$L_D^{latent} = \left\| D_L(z_U^g) - 0 \right\|_1 + \left\| D_L(z_U^r) - 1 \right\|_1, \tag{1}$$

$$L_G^{latent} = -log\left\| D_L(z_U^g) - 0 \right\|_1 - log\left\| D_L(z_U^r) - 1 \right\|_1. \tag{2}$$

where the output of D_L is a real value between 0 and 1 indicating the confidence in its decision, where 1 and 0 represent the real and generated image classes respectively. The value of L_G^{latent} is minimized when $D_L() = 0.5$, which means D_L cannot classify accurately according to z_U. On the other hand, minimizing L_D^{latent} results in precise predictions by D_L.

z_R attempt to provide related information for image manipulation. We assume that each real and generated image class follows the Gaussian mixture distribution in the space of z_R because this distribution facilitates the distinction between two classes in extractor training [41]. The loss function (3) is defined as follows:

$$\begin{aligned} L_{cls} =& \text{diff}(y, \text{r}) log(\mathcal{N}(z_R^y, \mu^{y_\text{r}}, \Sigma^\text{r}) p(y_\text{r})) \\ &+ \text{diff}(y, \text{g}) log(\mathcal{N}(z_R^y, \mu^{y_\text{g}}, \Sigma^\text{g}) p(y_g)). \end{aligned} \tag{3}$$

where diff() denotes a function returning -1 if the inputs are the same, otherwise, the output is $+1$. μ^y and Σ^y denote the mean and covariance of the class y in the feature space, and $p(y)$ represents the prior probability of y.

3.1.2. Generator

To evade image generation detection, our model aims to construct an image $\hat{\mathbf{x}}^g$ containing the semantic information of the generated image x_g and the characteristics of real images. Therefore, a residual image is generated by the generator G according to z_U^g extracted from x_g and z_R^r from any real image.

In order to increase the distribution similarity between z_U^g and the real image, z_R^r and z_U^g are first processed by the adaptive instance normalization module (AdaIN) [42] before sending to G. AdaIN is used to transfer style features to content features in image-style transfer applications. Similarly, our model treats z_R^r as a type of style that influences semantic features z_U^g before the image generation. Then, the distribution of z_U^g is normalized according to z_R^r by adjusting the mean and variance in AdaIN.

The objective of filter F is to reduce the high-frequency information in a generated image. This is crucial because high-frequency information can be exploited for generated image detection, especially because deconvolution operations in GANs often introduce

substantial high-frequency components into generated images [43]. To increase concealment, the high-frequency components of x^g are first removed and then combined with x^{res} to generate $\hat{\mathbf{x}}^g$. A generated image is split into 8×8 patches and Discrete Cosine Transform (DCT) obtains coefficient matrices of different frequencies from each patch by converting it into the frequency domain. The coefficients of different frequency components are divided by different values. The divisor of the high-frequency coefficient is larger than low-frequency coefficient. Finally, the coefficient matrix is inversely transformed into an image to obtain x_p^g. Compared with a traditional low-pass filter, the proposed method can be customized to reduce specific high-frequency components without the need for human eyes observation. Compared to JPEG compression, JPEG requires one quantization matrix for each Y and C channel in the YCrCb space, while F only needs a quantization matrix for all RGB channels. The time complexity of our model can be significantly reduced. After processing by the above modules, a camouflaged image is obtained by:

$$\hat{x}^g = x^{res} + x_p^g = G(E_U(x^g), E_R(x^r)) + x_p^g. \tag{4}$$

G is trained by minimizing the L1 loss between x^g and $\hat{x}^g$, and the L1 loss between z_R^r and $\hat{z}_R^g$ in (5) and (6) respectively. The reconstruction loss (5) ensures that the similarity between the manipulated image and its original image, and L_R loss (6) guarantees that z_R^r can still be extracted from the manipulated image to ensure consistency.

$$L_{rec} = \left\| \hat{x}^g - x^g \right\|_1, \tag{5}$$

$$L_R = \left\| \hat{z}_R^g - z_R^r \right\|_1 = \left\| E_R(\hat{x}^g) - z_R^r \right\|_1. \tag{6}$$

3.1.3. Discriminator

The robustness of the $\hat{x}^g$ against a generated image detector is enhanced by another discriminator D, which identifies whether an image has been generated. The adversarial training losses of the discriminator and generator are formulated as (7) and (8).

$$L_D = -log(1 - D(\hat{x}^g)) - log(D(x^r)), \tag{7}$$

$$L_G = -log(D(\hat{x}^g)). \tag{8}$$

3.2. Training phase

E_R, μ^y and Σ^y is first trained by minimizing L_{cls} to obtain separable distributions of $\mathcal{N}(\mu^r, \Sigma^r)$ and $\mathcal{N}(\mu^g, \Sigma^g)$. Afterward, all E_U, E_R, G, D, and D_L are trained jointly. G and E_R minimize $L_{E_R,G}$ as (9) while E_U is trained with L_{E_U} as (10). D and D_L minimize L_D and L_D^{latent} respectively.

$$L_{E_R,G} = \lambda_{rec} L_{rec} + \lambda_G L_G + \lambda_R L_R. \tag{9}$$

$$L_{E_U} = L_{E_R,G} + \lambda_G^{latent} L_G^{latent}. \tag{10}$$

3.3. Generation phase

To manipulate the generated image x^g, x^{res} is generated according to z_U^g extracted from x^g and z_R^r relying on a real image. However, a real image does not exist during the manipulation process. As a result, z_R^r is replaced by the mean of the real-related features of the real images used during training. $\hat{x}^g$ is calculated as:

$$\hat{x}^g = x^{res} + x_p^g = G(E_U(x^g), \mu^r) + x_p^g. \tag{11}$$

4. Experiments

The proposed model is evaluated and compared with State-Of-The-Art (SOTA) methods in different scenarios. An ablation experiment is carried out to verify the importance of each component in our proposed model.

4.1. Experimental settings

GAN-based methods including *ProGAN* [19], *SNGAN* [20], *MMDGAN* [22], *CramerGAN* [21], and *StyleGAN3* [44] are considered in our study. 10,000 real images from CELEBA, LSUN-bedroom, and FFHQ, respectively, and 10,000 pre-generated images of the GAN models provided in their studies are randomly selected to form a training set for the evasion methods. The parameters of our method are determined by conducting preliminary experiments with the aim of maximizing the performance of our method. As a result, the parameters are set as: $\lambda_G = \lambda_D = \lambda_G^{latent} = 10$, $\lambda_R = 1$ and $\lambda_{rec} = 10^3$. The number of training epochs in the first and second stages are 10 and 40, respectively. The source code of our method will be shared in GitHub once the paper is accepted.

In order to evaluate the evasion ability, 1,000 pre-generated images of the GAN models are randomly selected as a test set, ensuring that these images are distinct from those used in the training set. The CNN-based Detector (*CNND*) [6], GAN DCT Analysis (*DCTA*) [7], GAN FingerPrint Analysis (*FPA*) [10], and Co-Occurrence Matrices (*COM*) [27] are chosen as representatives of the pixel-based, frequency-based, fingerprint-based, and basic attributes-based methods to evaluate the evasion ability. All detectors except

Table 1
Average performance (Mean ± Stand Deviation, %) of evading different image generation detections by manipulating the images generated by *ProGAN*, *SNGAN*, *MMDGAN*, and *CramerGAN* in Celeba. 'Baseline' indicates the performance of using original images generated by the GAN model. All values are rounded to two decimal places, except COSS which is rounded to three. ↓ (↑) indicates a smaller (larger) value is preferable in a criterion. The best performance among all three evasion methods is bolded in each criterion.

CELEBA	CNND(%) ↓	DCTA(%) ↓	FPA(%) ↓	COM(%) ↓	FID↓	COSS↑
Baseline	86.60±17.16	99.88±0.13	99.98±0.05	98.28±1	27.89±9.45	100.000±0.000
PCA [40]	65.93±16.57	74.03±26.41	67.08±28.76	46.98±14.58	37.66±5.65	99.982±0.001
KSVD [40]	72.55±13.72	70.73±22.30	67.50±18.66	56.3±14.37	36.51±6.90	**99.984**±0.005
Ours	**65.65**±14.53	**13.60**±4.02	**36.18**±23.01	**7.6**±4.32	**34.97**±2.64	99.982±0.001

Table 2
Average performance (Mean ± Stand Deviation, %) of evading different image generation detections by manipulating the images generated by *ProGAN*, *SNGAN*, *MMDGAN*, and *CramerGAN* in LSUN-bedroom. 'Baseline' indicates the performance of using original images generated by the GAN model. All values are rounded to two decimal places, except COSS which is rounded to three. ↓ (↑) indicate a smaller (larger) value is preferable in a criterion. The best performance among all three evasion methods is bolded in each criterion.

LSUN-bedroom	CNND(%) ↓	DCTA(%) ↓	FPA(%) ↓	COM(%) ↓	FID↓	COSS↑
Baseline	99.06±1.45	99.96±0.08	99.74±0.19	99.44±0.98	29.38±23.17	100.000±0.000
PCA [40]	**91.5**±4.45	98.34±1.14	93.13±3.07	96.21±2.84	60.12±30.99	99.955±0.005
KSVD [40]	93.84±2.68	98.39±1.07	93.37±2.88	80.50±21.00	46.82±19.99	**99.988**±0.001
Ours	93.63±0.67	**97.87**±1.47	**46.70**±23.30	**71.83**±35.30	**40.27**±14.77	99.976±0.004

CNND are trained specifically for images generated by a particular GAN model. *CNND*, which is trained according to *ProGAN*, can be generalized to other GAN models [6]. *CNND*, *DCTA*, and *FPA* are implemented using all settings provided by their respective authors, while *COM* is trained by us following closely outlined procedures in [27]. A *COM* detector is trained for each GAN method using a training set containing 18,151 randomly selected images from CELEBA as real images, and the same number of images are generated by each GAN method as the generated images.

The evasion performance is evaluated in terms of detection accuracy, degree of image changes, and authenticity of the edited image. The COSS score [45], which calculates the cosine similarity between clean and adversarial samples, is utilized to measure the degree of image changes. The value range is [0,100]. A score closer to 100 signifies that the adversarial sample is more similar to the clean sample. A notable benefit of the COSS score is its sensitivity in detecting subtle modifications in images. Frechet Inception Distance (FID) [46] is used to measure the authenticity of the edited image. FID calculates the distance between real and GAN images in the feature space mapped by the Inception network. A lower FID score means that the GAN image is more authentic and diverse.

4.2. Results and discussion

4.2.1. Evasion of detection

Three State-Of-The-Art (SOTA) methods for evading image generation detection, including *PCA*, *KSVD* [40], and BAttGAND [47] are compared with our proposed method. The training settings and all parameters closely follow the studies. The training set is the same as that used in our method. The average performances of the methods on camouflaged samples generated by *ProGAN*, *SNGAN*, *MMDGAN*, and *CramerGAN* in Celeba and LSUN-bedroom are shown in Tables 1 and 2 respectively.

The results shown in Table 1 indicate that all general evasion methods reduce the accuracy of all detection methods by sacrificing the quality of images in CELEBA, *i.e.,* all FID values increase and COSS values decrease. Our method dramatically reduces all detection rates by 20.95%, 86.58%, 63.80%, and 90.68%, and is significantly better than the other evasion methods, especially under the detection of DCTA, FPA, and COM. Moreover, the quality of our edited images is also the best, *i.e.,* FID is the smallest among all methods, although our edited images are not the closest to the original ones, *i.e.,* COSS is not the smallest. It can be clearly observed that although the COSS of our method is 0.002% less than that of KSVD, it reduces the success rate of the four detectors by an average of 36.01% more than that of KSVD. Thus means small additional changes can lead to much greater performance improvements.

As a residual image is added to its original image, rather than generating a complete new image in our model, the original quality can be maintained. Our method requires the edited image to look similar to the original image, which is not considered by other methods. Moreover, the small variances suggest that the performance of our method is the most stable among all GAN models, particularly for DCTA, COM, FID, and COSS. *PCA* and *KSVD* have similar attack effects. It may be because they rely on dimensionality reduction and reconstruction methods. A GAN-generated image is reconstructed according to the mapping of real images to remove the characteristics of fake images. The image processed by *KSVD* is in better quality. However, it is the most unstable on various GANs. In terms of attack success rate, *PCA* outperforms *KSVD* in all cases except DCTA.

The results of all methods in LSUN-bedroom shown in Table 2 are similar to those of CelebA. Generally, the quality of the generated images in LSUN-bedroom is worse than that in CelebA, *i.e.,* FID scores of the baseline in LSUN-bedroom are higher. This indicates that larger modifications are required, and the methods are inferior to CelebA in terms of the attack effect. Our method obtains the lowest FID values among all methods, which indicates that the image quality of our method is the best. In cases that the modification

Table 3
Performance of evading different image generation detections by manipulating the images generated by *StyleGAN3* in FFHQ. 'Baseline' indicates the performance of using original images generated by the GAN model. All values are rounded to two decimal places, except COSS which is rounded to three. ↓ (↑) indicate a smaller (larger) value is preferable in a criterion. The best performance among all three evasion methods is bolded in each criterion.

FFHQ	CNND(%) ↓	DCTA(%) ↓	FPA(%) ↓	COM(%) ↓	UFD(%) ↓	FID↓	COSS↑
Baseline	70.00	95.80	99.00	95.30	76.20	28.05	100.000
PCA [40]	19.20	87.30	88.90	95.00	76.10	39.89	99.979
KSVD [40]	60.90	92.20	89.00	46.40	75.60	49.27	**99.996**
BAttGAND [47]	22.20	**65.50**	84.50	**25.10**	87.50	**37.57**	99.969
Ours	**5.10**	79.30	**78.90**	25.60	**57.10**	38.51	99.964

of our method is smaller than PCA, our method still has a stronger attack effect. KSVD is the method with the least modification, but its FID is not as good as that of our method. Moreover, our method achieves the best results in three out of the four detectors, *i.e.,* 2.15% in DCTA, 53.04% in FPA, and 27.61% in COM, and performs similarly to other methods in LSUN-bedroom.

Our proposed model is subjected to additional empirical testing in various settings using other recent methods. Additional attack and detection methods, BAttGAND [47] and UniversalFakeDetect (UFD) [39], are considered in the dataset FFHQ [48]. The images are generated StyleGAN3 [44].

Consistent with prior experimental outcomes, our approach exhibits enhanced efficacy across a broad spectrum of detectors, as depicted in Table 3. More precisely, UFD only achieves a 57.10% success rate in identifying our method, equating this performance nearly to that of a random guess. This level of accuracy is markedly inferior to that achieved when detecting the other three attack strategies, each of which achieves an accuracy rate exceeding 75%. The performance of BAttGAND is comparable to our method, with BAttGAND outperforming our method by 14.20% and 0.50% in detections using DCTA and COM, respectively. However, it significantly underperforms under UFD with an accuracy of 87.50%. A plausible explanation for this could be that BAttGAND focuses solely on the characteristics of simulated real images rather than actual real images. The distinction between real and simulated images introduces discriminative features that facilitate differentiation between genuine and counterfeit images. While PCA and KSVD yield commendable results with CNND and COM detectors, respectively, they do not match the effectiveness of our method against other detectors. These findings underscore the resilience of our technique in the face of various types of generated images and arrays of detectors.

The performance of our method is also discussed subjectively. Fig. 3 illustrates an example of original images generated by a GAN model, their camouflaged image, and the corresponding modified content image. As the modification of the images processed by all methods is not obvious in the sense of human vision, each pixel value x in modified content images is adjusted as a new value x' by $x' = 255 - 3x$ for better illustration.

All methods edit the image roughly according to the outline of a face, but our method focuses on the outline more than others, *i.e.,* our modification images demonstrate a more obvious face outline. On the other side, *PCA* and *KSVD* modify many background pixels shown in the first and fourth samples of face images, while our method only considers on the content of a face and only slightly changes the background. This may be because our model learns the most significant part of the image for generated image detection during learning. Only pixels that make the image closer to real images are edited in our method.

4.2.2. Image modification ratio

The image quality and evasion ability are inversely proportional, *i.e.,* a detection accuracy is higher on an image with small manipulation and vice versa. This section investigates how the ratio of image modification affects the balance between evasion ability and image quality using ProGAN in CELEBA as an example. An image with varying degrees of modification is obtained by adding $r \times \Delta x^g$ to the original image, where r represents the degree of image modification, and Δx^g denotes the difference between the camouflaged and original images. Fig. 4 shows the influence of r on different criteria. The x- and y-axis denote the ratio and criterion, respectively. The blue, green, and red lines represent *PCA*, *KSVD*, and our method, respectively.

The detection accuracy and image quality decrease with increasing of the ratio for all evasion methods in general. The observations in this experiment comprise the findings in the previous section. Basically, the ranking of all methods is consistent with different ratios. However, the detection accuracy of our method decreases rapidly with the increase of r, especially in DCTA after 40%, FPA after 30%, and COM after 50%. Compared to our method, the detection rates for the images processed by *PCA* and *KSVD* drop more mildly. However, the trends of the quality drop of the processed images are similar for all methods.

The efficiency of the tradeoff between image quality and evasion ability is further illustrated in Fig. 5. The y-axis represents the average detection accuracy of CNND, DCTA, FPA, and COM, while the x-axis indicates the image quality in terms of FID and COSS. Each method contains 10 points representing the results when $r = 10\%, 20\%, ..., 100\%$. When the image quality is high, no significant difference is observed in the detection rates among the methods. However, the average detection accuracy of our method drops dramatically when image quality is sacrificed compared to the other two methods, *i.e.,* The slope of the curses of *PCA* and *KSVD* is much smaller than our method. The results confirm that our method works more efficiently.

4.2.3. Ablation study

The importance of each component of our model is evaluated in this section. Our full model is compared to the models with missing components by using ProGAN on CELEBA, including D, D_L, L_R, and F. The results are shown in Table 4.

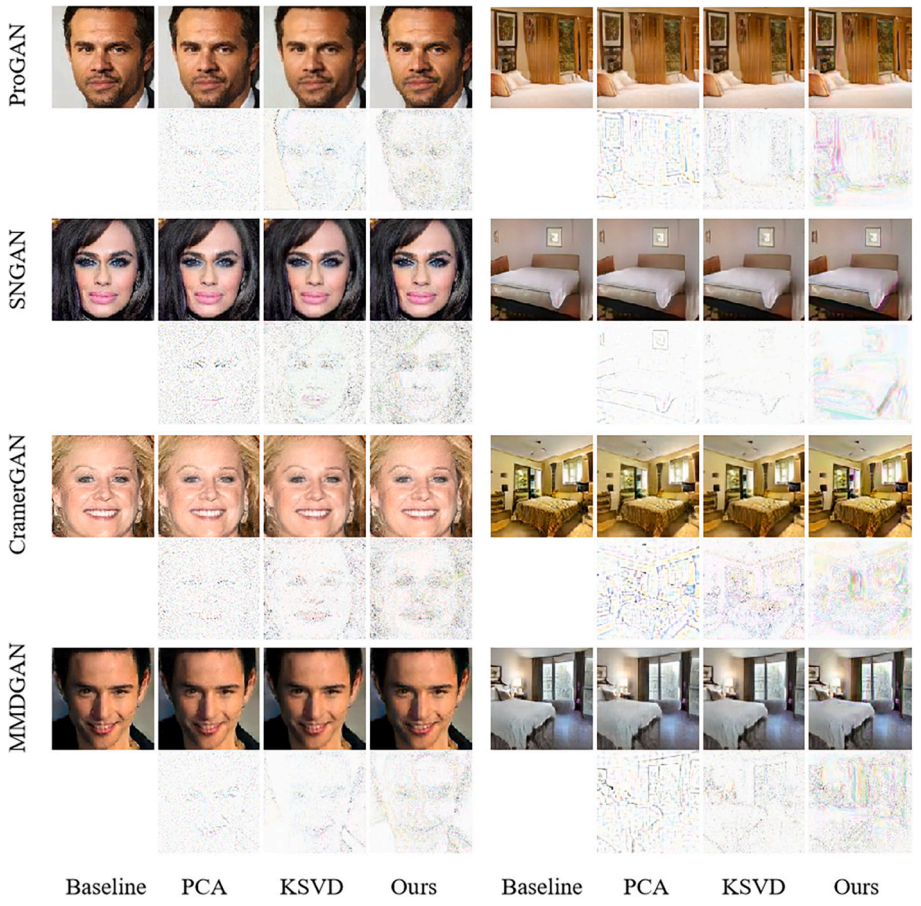

Fig. 3. Examples of original images generated by a GAN model (Baseline), their camouflaged images generated by *PCA*, *KSVD* and our method, and the difference between the original and camouflaged images. The difference is amplified for better illustration.

Table 4
Evasion performance (Mean ± Stand Deviation, %) of our proposed model without a component on *ProGAN*, *SNGAN*, *MMDGAN*, and *CramerGAN* in CELEBA. All values are rounded to two decimal place, except COSS which is rounded to three. ↓ (↑) indicate a smaller (larger) value is preferable in a criterion. A bolded value indicates the best.

CELEBA	CNND(%) ↓	DCTA(%) ↓	FPA(%) ↓	COM(%) ↓	FID↓	COSS↑
Ours	**65.65**±14.53	**13.60**±4.02	36.18±23.01	**7.6**±4.32	34.97±2.64	99.982±0.001
Ours w/o D	83.90±11.67	25.50±4.55	74.25±33.22	15.80±8.58	35.53±2.07	99.990±0.001
Ours w/o D_L	79.53±12.43	21.68±5.14	67.20±31.91	12.98±6.53	35.37±2.10	99.987±0.001
Ours w/o L_R	78.40±13.06	15.10±5.01	**33.83**±21.31	7.73±4.19	35.85±2.01	99.978±0.001
Ours w/o F	77.10±4.08	65.63±1.15	45.75±2.83	38.68±1.30	**29.63**±1.32	**99.998**±0.000

Our full model outperforms all the methods in terms of evasion ability, except for the detection success rate of FPA, in which our model without L_R performs 2.25% better than our full model. L_R encourages the same real-related information as the original and manipulated images obtained from E_R; thus, it plays an insignificant role in our model. That why the performance of our model without L_R is closest to that of the full model, although the image quality is worse.

On the other hand, our model without D performs the worst, *i.e.,* it plays an important role in our model. Without D, the adversarial feedback information to the generator is reduced, which prevents the generator learn properly. The results also suggest that D_L is an important component of our model. D_L reduces related information from z_U, which provides useful information for generating a residual image with the same structure as the original image, and unnecessary changes to manipulated images can be avoided. The image quality of Ours w/o F is the best among all methods, but its corresponding detection success rate is significantly higher for DCTA and COM. This indicates that although F contributes to the attack, it is not dominant over the entire process.

4.2.4. Evasion with prior information

We assume that the detection method was completely unknown in previous experiments. One flexibility of our model is the adaption of prior information, *i.e.,* the performance of our model can be improved by considering the prior information of the

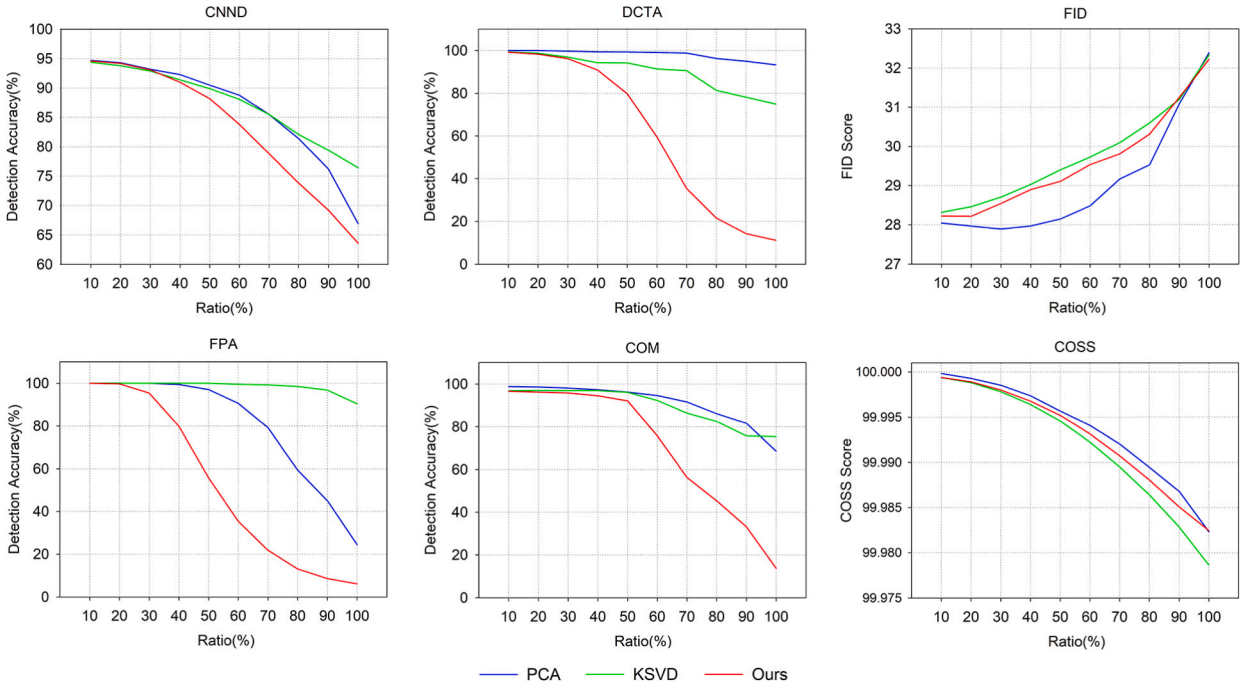

Fig. 4. Performance comparison among *PCA*, *KSVD*, and our method using different ratios of image modification.

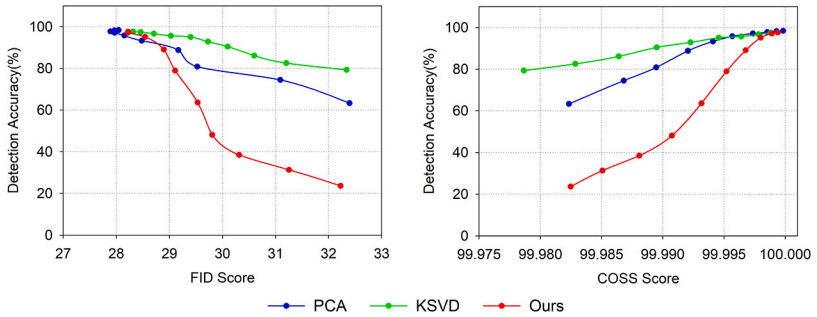

Fig. 5. The tradeoff between average detection accuracy on the ProGAN and the image quality of *PCA*, *KSVD*, and our method.

detector. DCTA, which considers frequency domain features, is used as an example for demonstration. Assume the adversary knows that the DCTA detector is selected. During training, in addition to the original discriminator D, DCTA is also considered in order to improve the evasion ability of the image $\hat{x}^g$ processed by our model. D and DCTA are trained in the same manner, and their losses jointly affect the generation of residual images. Two detector-dependent evasion methods, including $CW\text{-}L_2$ (CW-DCTA) [49] and transfer attack (TA-DCTA) [24] which are white- and black-box attack methods, respectively, are also considered. Both methods are trained against DCTA, and their training settings closely followed those in their studies.

The experimental results obtained on ProGAN are shown in Table 5. Our method adapting DCTA (Ours-DCTA) achieves better results than our standard model in terms of evasion. The identification accuracies of all detection methods on the images processed by Ours-DCTA are the lowest among all methods, except DCTA. As the original D considers the pixel domain while the frequency domain feature is focused by DCTA, these two discriminators may complement each other in order to increase the difficulty of identification. However, Ours-DCTA requires more image modification than Ours, *i.e.,* FID is higher and COSS is lower. Although CW-DCTA achieves the best performance on DCTA, *i.e.,* 3.8% only, its processed images can be detected easily by other detectors. TA-DCTA only changes the images slightly, which may explain why its evasion performance is undesirable. These results confirm that our method can easily adapt the prior knowledge of the detectors. The improvement is not limited to only the chosen detector but also other detectors.

4.2.5. Impact of encoded feature dimension

Our attack strategy replies on utilizing a residual image. Its dimension plays a pivotal role in influencing performance because it dictates the volume of information used in the attack. The purpose of this section is to examine how the dimension of the encoded features affects our method's effectiveness. This exploration will offer valuable insights into designing the encoder structure and

Table 5
Performance (%) of evasion methods including general evasion method (*i.e.,* our method, *PCA*, *KSVD*), and specifically designed evasion method for DCTA (*i.e.,* our method, CW-L2 and TA) on *ProGAN* in CELEBA. All values are rounded to one decimal place, except COSS which is rounded to two. ↓ (↑) indicates a smaller (larger) value is preferable in a criterion. The best performance among all evasion methods is bolded in each criterion.

CELEBA	CNND(%) ↓	DCTA(%) ↓	FPA(%) ↓	COM(%) ↓	FID↓	COSS↑
ProGAN [19]	94.5	99.9	99.9	99.0	16.7	100.00
PCA [40]	66.7	93.3	24.4	68.4	32.4	99.98
KSVD [40]	76.4	75.0	90.4	75.4	32.3	99.98
Ours	63.6	11.2	6.1	13.7	32.2	99.98
CW-DCTA [49]	79.4	**3.8**	62.7	38.7	**31.0**	99.99
TA-DCTA [24]	91.2	59.7	99.9	95.7	31.8	**100.00**
Ours-DCTA	**44.7**	8.5	**1.1**	**3.3**	36.6	99.93

Table 6
Average performance of evading different detectors with different related and unrelated feature dimensions. The horizontal number represents the dimension of the unrelated features. The format is [# of channels, width, height]. The vertical number is the dimension of the related feature, *i.e.,* related to the label. So it is a one-dimensional tensor. The performance is measured by the detection success rate of the detector, so a smaller value is preferable. The darker the color, the better the performance.

	[32,128,128]	[64,64,64]	[128,32,32]	[256,16,16]	[512,8,8]
128	30.58	24.00	42.00	65.80	44.25
256	32.70	21.15	51.38	49.48	65.65
512	29.38	12.85	36.03	54.18	76.48
1024	29.18	22.70	41.28	64.20	61.90

Table 7
COSS scores of our method with different related and unrelated feature dimension. The horizontal number represents the dimension of the unrelated features. The format is [# of channels, width, height]. The vertical number is the dimension of the related feature, *i.e.,* related to the label. So it is a one-dimensional tensor. The darker the color, the better the performance.

	[32,128,128]	[64,64,64]	[128,32,32]	[256,16,16]	[512,8,8]
128	99.960	99.960	99.978	99.985	99.988
256	99.984	99.983	99.957	99.985	99.948
512	99.591	99.591	99.979	99.909	99.981
1024	99.804	99.804	99.968	99.805	99.914

Table 8
FID scores of our method with different related and unrelated feature dimension. The horizontal number represents the dimension of the unrelated features. The format is [# of channels, width, height]. The vertical number is the dimension of the related feature, *i.e.,* related to the label. So it is a one-dimensional tensor. The darker the color, the better the performance.

	[32,128,128]	[64,64,64]	[128,32,32]	[256,16,16]	[512,8,8]
128	33.466	32.466	31.931	33.465	32.994
256	33.099	32.007	32.657	34.061	32.580
512	34.500	33.500	33.577	33.153	34.389
1024	33.949	33.949	34.933	34.443	35.660

guides us in optimizing our attack methodology to realize better performance. This section presents images from the CelebA dataset. ProGAN is used to generate images for training and testing.

Table 6 shows our method's average performance (measured with the detection success rate of detectors) in evading different detectors with different related and unrelated feature dimensions. The horizontal number represents the dimension of unrelated features. The format is [number of channels, width, height]. The vertical number is the dimension of the related feature, *i.e.,* related to the label. Thus, it is a one-dimensional tensor. The average detection rate across all detectors is lowest consistently under all configurations when the setting $[64, 64, 64]$ is employed. Images camouflaged with these parameters are elusive and challenging to detect. In addition to stealthiness, the quality of the camouflaged images is also an important factor to consider. The COSS and FID scores, which evaluate the quality of images under various attack settings, are presented in Tables 7 and 8. The results in the COSS values indicate that the quality of stealthier images tends to be lower than that of less stealthy images. Specifically, $[256, 16, 16]$ and $[512, 8, 8]$ yield the highest-quality images despite their poor stealth performance. However, generally, $[64, 64, 64]$ also achieves the

Table 9
The average detection rates of our models, which were trained using images generated by various GAN models, against CNND, DCTA, FPA, and COM. Each column and row specify the GAN models involved in the training and test phases respectively.

	ProGAN	SNGAN	CramerGAN	MMDGAN
ProGAN	**22.425**	48.225	46.375	38.025
SNGAN	36.300	**35.200**	55.975	51.825
CramerGAN	27.175	41.325	**40.575**	36.950
MMDGAN	26.675	37.650	44.200	**35.600**

Table 10
The detection rates of CNNDs trained by images crafted by different attack methods. Each column and row represent the attack method involved in the training and test phases respectively. 'Clean' refers to the inclusion of only clean images.

	clean	KSVD	PCA	BattGAND	Ours
KSVD	19.2	**90.1**	3.3	40.7	1.9
PCA	60.9	21.1	**97.4**	88.3	2.9
BattGAND	22.2	5.4	34.5	**94.1**	11.3
Ours	5.1	0.2	11.1	1.2	**83.3**

best performance in terms of FID. When selecting appropriate settings, we recommend prioritizing the stealthiness of camouflaged images before considering their quality. Therefore, a dimension of 256 and the setting $[64, 64, 64]$ are selected. This choice is underpinned by the fact that, within the $[64, 64, 64]$ column, images with a dimension of 256 demonstrate the best quality, as evidenced by their highest COSS value and lowest FID value.

4.2.6. Transferability between GANs

This section assesses the transferability of our method's performance across images produced by different GAN models. Our method undergoes training with real images and images generated by one of several GAN models, including ProGAN, SNGAN, CramerGAN, and MMDGAN. The CelebA dataset is used in this evaluation. The following training, the model is deployed to camouflage images generated by the above GAN models. Table 9 displays the average detection success rates for CNND, DCTA, FPA, and COM. The table is organized such that each row and column is associated with a GAN model used in the test and training sets.

The models exhibit optimal performance when both training and test images are generated from the same GAN model for all scenarios. The experimental results demonstrate that attacks on images generated by ProGAN can be advantageous relative to camouflaging other GAN models. The performance of our model trained with ProGAN is the best among all GAN models used in our experiments. Despite this, models trained using images from CramerGAN and MMDGAN show that test images originally from these GAN models are less stealthy than those generated by ProGAN. However, the converse is true for ProGAN; it does not significantly enhance our model's ability to camouflage images from other GANs, as evidenced by the relatively high detection rates for non-ProGAN generated images in the ProGAN row. The data for both column and row associated with CramerGAN are relatively high, which indicates a distinct performance gap between CramerGAN and other GAN models. This suggests that the insights obtained from CramerGAN are not well transferable to other models. In conclusion, the results demonstrate significant variance in the characteristics of images produced by different GAN models. For practical applications, it is recommended that our model undergoes specific training with images from the target GAN model to realize the best performance. When no specific generation method is targeted, ProGAN is the preferred option due to its superior transferability across a range of GAN models.

4.2.7. Evasion on defender trained with camouflaged images

In the previous sections, experiments were conducted under the assumption that the defender had no prior knowledge of the attack, resulting in the training of detectors on uncamouflaged generated images. This section investigates the performance of attack methods when the defender is provided with knowledge of the attack. In particular, the detection model is trained using generated images camouflaged by one of the attack methods. CNND is used as the detector. The detection rates in the FFHQ dataset are displayed in Table 10, where each row and column correspond to the attack method used in the training and testing images, respectively. The term "Clean" refers to a detector trained on clean images.

The results clearly show that the highest detection rates are observed along the diagonal direction, indicating a strong correlation between the training and test datasets. In scenarios where the detector is trained using samples camouflaged by the actual attack method, it poses the most challenging situation for the adversary. Although all attack methods perform worse in these scenarios, our method exhibits the lowest detection success rate. This suggests that the samples camouflaged by our method exhibit significant variation, making it difficult for the detector to learn their patterns. Furthermore, our method demonstrates the least correlation with other attack methods. As seen in the table, models tested on KSVD, PCA, and BattGAND achieve detection success rates of only 1.9%, 2.9%, and 11.3%, respectively, when trained on our camouflaged samples.

Table 11
Detection rates of CNNDs trained on images processed with JPEG compression and Gaussian blur. A lower quality value indicates a higher degree of compression in JPEG compression, resulting in poorer image quality after compression. The original image quality is typically denoted as 1.00. On the other hand, in Gaussian blur, a larger sigma value signifies the presence of blurrier images.

	No Preprocessing	Pre-process by JPEG compression			Pre-process by Gaussian blur		
		Quality 0.80	Quality 0.65	Quality 0.30	Sigma 1	Sigma 2	Sigma 4
baseline	70.0	21.8	16.2	17.4	68.6	64.4	66.8
ours	5.1	6.4	8.1	7.4	3.3	5.8	8.6

4.2.8. Influence on image processing

This section aims to explore the potential advantages of utilizing image processing techniques in detecting our attack method. Experiments to examine the impact of training set modifications through image processing on the detection performance are carried out. Specifically, CNND is trained using a combination of real images and ProGAN-generated images. The training data undergoes pre-processing using JPEG compression and Gaussian blur with various parameters. Table 11 presents the detection rates of CNNDs trained on images processed with JPEG compression and Gaussian blur. The test images used for evaluation are generated from both the StyleGAN3 (baseline) and our proposed method.

The results suggest that there is a slight improvement in the accuracy of detecting generated images concealed by our method when pre-processing is applied using JPEG compression. However, it is observed that the detection rate remains below 9%, indicating the robustness of our method under this scenario. A similar conclusion can be drawn when Gaussian blur is used as a pre-processing technique. On the other hand, the baseline method experiences a significant decrease in detection rates after pre-processing. Therefore, the impact of image pre-processing on detection is inconclusive.

5. Conclusions

This paper introduces a sophisticated method to evade general detection systems for GAN-generated images. It achieves this by disentangling image features into those that are related and unrelated to the identification between real and GAN-generated images. In the process of camouflaging a generated image, our method employs unrelated features from the generated image along with related features derived from real images. Unrelated features are utilized to capture the essential content information of the generated image, whereas related features from real images are employed to enhance the characteristics of the camouflaged image, making it less detectable by conventional detection mechanisms.

The experimental findings of this study underscore the versatility and efficacy of our model in camouflaging images generated by the latest GAN models, making it a powerful tool against most current detection techniques. Our model demonstrates superior performance over SOTA methods in circumventing commonly employed detection methods by requiring minimal image modifications. Furthermore, we investigate the balance between the extent of image alteration and the capability to evade detection, and our results demonstrate our method's exceptional efficiency in this regard. Notably, our model exhibits the ability to leverage prior knowledge about detectors to further enhance its evasion proficiency across a broad spectrum of detection methods. This adaptability signifies a significant advancement in the development of more resilient camouflaging techniques. Moreover, we investigate the impact of varying the dimension of encoded features and performance across images generated by different GAN models, which provides valuable insights into optimizing our evasion method. In addition, the experimental results reveal a weak correlation between our method and other attack strategies, which suggests that the proposed method introduces a novel and distinct pathway to evade detection. This distinctiveness increases the stealth of generated images camouflaged by our method. The experiment results also indicate that our method successfully evades detection, even when considering image pre-processing during the training phase, which may not effectively improve detection rates.

One possible future research direction could involve exploring the use of disentangled related and unrelated features to develop an effective method to identify GAN-generated images. By understanding and harnessing these differentiated features, the space within the related features may be beneficial relative to accurately identifying camouflaged images. Moreover, the investigation could extend beyond image detection to include additional security-related applications, for instance, face liveness detection. By applying insights gained from the study of disentangled features in GAN-generated images, future work can develop more sophisticated systems capable of detecting spoofing attacks in biometric security, thereby bolstering the effectiveness of liveness detection technologies.

CRediT authorship contribution statement

Patrick P.K. Chan: Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Chuanxin Zhang:** Writing – original draft, Software, Methodology, Formal analysis. **Haitao Chen:** Writing – review & editing, Software, Conceptualization. **Jingwen Deng:** Writing – review & editing, Writing – original draft, Validation, Software. **Xiao Meng:** Writing – original draft, Validation, Software. **Daniel S. Yeung:** Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[2] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan, in: Proc. CVPR, 2020.
[3] Z. He, W. Zuo, M. Kan, S. Shan, X. Chen, Attgan: facial attribute editing by only changing what you want, IEEE Trans. Image Process. 28 (2019) 5464–5478, https://doi.org/10.1109/TIP.2019.2916751.
[4] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: CVPR, 2017.
[5] This person does not exist, https://thispersondoesnotexist.com/, 2022.
[6] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, A.A. Efros, Cnn-generated images are surprisingly easy to spot... for now, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[7] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, T. Holz, Leveraging frequency analysis for deep fake image recognition, in: Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 3247–3258, http://proceedings.mlr.press/v119/frank20a.html.
[8] K. Chandrasegaran, N.-T. Tran, N.-M. Cheung, A closer look at Fourier spectrum discrepancies for cnn-generated images detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7200–7209.
[9] C. Dong, A. Kumar, E. Liu, Think twice before detecting gan-generated fake images from their spectral domain imprints, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7865–7874.
[10] N. Yu, L.S. Davis, M. Fritz, Attributing fake images to gans: learning and analyzing gan fingerprints, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
[11] S. Agarwal, T. El-Gaaly, H. Farid, S.-N. Lim, Detecting deep-fake videos from appearance and behavior, in: 2020 IEEE International Workshop on Information Forensics and Security (WIFS), 2020, pp. 1–6.
[12] X. Yang, Y. Li, S. Lyu, Exposing deep fakes using inconsistent head poses, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 8261–8265.
[13] U. Ciftci, I. Demir, L. Yin, Fakecatcher: detection of synthetic portrait videos using biological signals, IEEE Trans. Pattern Anal. Mach. Intell. (2020) 1, https://doi.org/10.1109/TPAMI.2020.3009287.
[14] T. Jung, S. Kim, K. Kim, Deepvision: deepfakes detection using human eye blinking pattern, IEEE Access 8 (2020) 83144–83154, https://doi.org/10.1109/ACCESS.2020.2988660.
[15] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, H. Li, Protecting world leaders against deep fakes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.
[16] S. Jandial, P. Mangla, S. Varshney, V. Balasubramanian, Advgan++: harnessing latent layers for adversary generation, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019, pp. 2045–2048.
[17] H. Zhang, B. Chen, J. Wang, G. Zhao, A local perturbation generation method for gan-generated face anti-forensics, IEEE Trans. Circuits Syst. Video Technol. 33 (2022) 661–676.
[18] Twitter, Synthetic and manipulated media policy, https://help.twitter.com/en/rules-and-policies/manipulated-media, 2023.
[19] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, arXiv preprint, arXiv:1710.10196, 2017.
[20] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, in: International Conference on Learning Representations, 2018.
[21] M.G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, R. Munos, The Cramer distance as a solution to biased Wasserstein gradients, arXiv:1705.10743, 2017.
[22] M. Binkowski, D.J. Sutherland, M. Arbel, A. Gretton, Demystifying mmd gans, arXiv:1801.01401 [stat.ML], 2018.
[23] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, T. Aila, Alias-free generative adversarial networks, in: Proc. NeurIPS, 2021.
[24] N. Carlini, H. Farid, Evading deepfake-image detectors with white- and black-box attacks, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2804–2813.
[25] S. Tariq, S. Lee, H. Kim, Y. Shin, S. Woo, Detecting Both Machine and Human Created Fake Face Images in the Wild, 2018, pp. 81–87.
[26] Y. Bai, Y. Guo, J. Wei, L. Lu, R. Wang, Y. Wang, Fake generated painting detection via frequency analysis, in: 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 1256–1260.
[27] L. Nataraj, T.M. Mohammed, B. Manjunath, S. Chandrasekaran, A. Flenner, M.J. Bappy, A. Roy-Chowdhury, Detecting gan generated fake images using co-occurrence matrices, Electron. Imaging 2019 (2019) 532-1, https://doi.org/10.2352/ISSN.2470-1173.2019.5.MWSF-532.
[28] S. McCloskey, M. Albright, Detecting gan-generated imagery using saturation cues, in: 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 4584–4588.
[29] P.P.K. Chan, Y. Wang, D.S. Yeung, Adversarial attack against deep reinforcement learning with static reward impact map, in: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, ASIA CCS '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 334–343.
[30] K. Chen, P.P.K. Chan, D.S. Yeung, Shilling attack detection using rated item correlation for collaborative filtering, in: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018, pp. 3553–3558.
[31] J. Zhang, W. Zhang, J. Huang, Hao Xie, Jiangqun Ni, Evading generated-image detectors: a deep dithering approach, Signal Process. 197 (2022), https://doi.org/10.1016/j.sigpro.2022.108558.
[32] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, B. Dong, You only propagate once: accelerating adversarial training via maximal principle, arXiv preprint, arXiv:1905.00877, 2019.
[33] M. Li, C. Deng, T. Li, J. Yan, X. Gao, H. Huang, Towards transferable targeted attack, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 638–646.
[34] J. Wang, G. Zhao, Haitao Zhang, Beijing Chen, A local perturbation generation method for gan-generated face anti-forensics, IEEE Trans. Circuits Syst. Video Technol. 33 (2023) 661–676, https://doi.org/10.1109/TCSVT.2022.3207310.

[35] T. Maho, T. Furon, E. Le Merrer, Surfree: a fast surrogate-free black-box attack, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10425–10434.
[36] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, in: Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, Chemom. Intell. Lab. Syst. 2 (1987) 37–52, https://doi.org/10.1016/0169-7439(87)80084-9, https://www.sciencedirect.com/science/article/pii/0169743987800849.
[37] M. Aharon, M. Elad, A. Bruckstein, K-svd: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (2006) 4311–4322, https://doi.org/10.1109/TSP.2006.881199.
[38] J.C. Neves, R. Tolosana, R. Vera-Rodriguez, V. Lopes, H. Proença, J. Fierrez, Ganprintr: improved fakes and evaluation of the state of the art in face manipulation detection, IEEE J. Sel. Top. Signal Process. 14 (2020) 1038–1048.
[39] C. Liu, H. Chen, T. Zhu, J. Zhang, W. Zhou, Making deepfakes more spurious: evading deep face forgery detection via trace removal attack, IEEE Trans. Dependable Secure Comput. (2023).
[40] Y. Huang, F. Juefei-Xu, R. Wang, Q. Guo, L. Ma, X. Xie, J. Li, W. Miao, Y. Liu, G. Pu, Fakepolisher: Making Deepfakes More Detection-Evasive by Shallow Reconstruction, 2020, pp. 1217–1226.
[41] W. Wan, Y. Zhong, T. Li, J. Chen, Rethinking feature distribution for loss functions in image classification, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 9117–9126.
[42] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: ICCV, 2017.
[43] R. Durall, M. Keuper, J. Keuper, Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7890–7899.
[44] T. Karras, M. Aittala, S. Laine, E. Harkonen, J. Hellsten, J. Lehtinen, T. Aila, Alias-Free Generative Adversarial Networks, vol. 2, 2021, pp. 852–863.
[45] V.K.P.-N. Tan, M. Steinbach, Introduction to Data Mining, vol. 4, Addison-Wesley, 2005.
[46] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local Nash equilibrium, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6629–6640.
[47] Z. Lou, G. Cao, M. Lin, Black-box attack against gan-generated image detector with contrastive perturbation, Eng. Appl. Artif. Intell. 124 (2023) 106594.
[48] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 4217–4228, https://doi.org/10.1109/TPAMI.2020.2970919.
[49] A. Gandhi, S. Jain, Adversarial perturbations fool deepfake detectors, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–8.