Generative Autoencoders as Watermark Attackers: Analyses of Vulnerabilities and Threats

Xuandong Zhao [*1] **Kexun Zhang** [*1] **Yu-Xiang Wang** [1] **Lei Li** [1]

Abstract

Invisible watermarks safeguard images' copyrights by embedding hidden messages detectable by owners. It also prevents people from misusing images, especially those generated by AI models. Malicious adversaries can violate these rights by removing the watermarks. In order to remove watermarks without damaging the visual quality, the adversary needs to erase them while retaining the essential information in the image. This is analogous to the encoding and decoding process of generative autoencoders, especially variational autoencoders (VAEs) and diffusion models. We propose a framework using generative autoencoders to remove invisible watermarks and test it using VAEs and diffusions. Our results reveal that, even without specific training, off-the-shelf Stable Diffusion effectively removes most watermarks, surpassing all current attackers. The result underscores the vulnerabilities in existing watermarking schemes and calls for more robust methods for copyright protection.

1. Introduction

Posting images online can be risky because malicious users may misuse them and violate the owners' copyright and privacy. Besides, advances in generative AI, such as DALLE-2, Imagen, and Stable Diffusion (Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022), can generate extremely photorealistic images which can mislead people into false beliefs (see examples in Figure 1). To protect images from misuse and achieve AI responsibility, major tech companies like Google are developing tools to trace the origin of images or identify synthetically generated content (Google, 2023; Wiggers, 2023). Invisible watermarks are one such tool that has been applied to embed secret messages detectable by the owner (Rombach et al., 2022). Typical watermarking schemes include bit manipulation (Wolfgang & Delp, 1996), signal processing (Ghazanfari et al., 2011; Holub & Fridrich, 2012) and learning-based methods (Zhang et al., 2019b;a; Fernandez et al., 2021).

Figure 1. AI-generated fake images from Twitter depicting the arrest of Donald Trump.

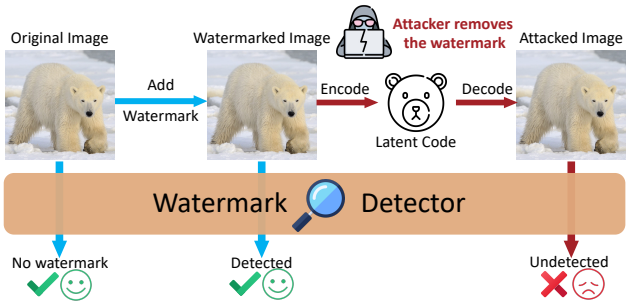

Figure 2. Remove invisible watermarks: Attackers first encode the watermarked image into a latent representation retaining essential information. They then decode to reconstruct the image and remove the watermark.

Adversaries can utilize various methods to remove the watermarks such as image transformations and image denoising (Dabov et al., 2007; Zhang et al., 2017; 2021; Hosam, 2019). While they can successfully erase weak watermarks, most of them cannot work well for more stronger watermarks, and they can severely hurt the image quality.

Recent studies on generative models have changed the situation. The process of watermark removal is equivalent to generating a new watermark-free image from the most essential information in the old. This is analogous to the encoding and decoding of generative autoencoders such as VAEs (Kingma & Welling, 2013) and diffusions (Ho et al., 2020). Motivated by the analogy, we propose to utilize generative autoencoders as watermark attackers. As shown in Figure 2, attackers can erase the watermark by first encoding

[*]Equal contribution [1]UC Santa Barbara. Correspondence to: Xuandong Zhao <xuandongzhao@ucsb.edu>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

the watermarked image to a latent code and then decoding it to a reconstructed image.

We evaluate our pipeline along with several baseline watermark removers. The results indicate that generative autoencoders, especially diffusions, can remove more invisible watermarks than most existing attackers while keeping the image quality intact. These results underscore the vulnerabilities of the existing watermark schemes and how generative models can pose threats to copyrights and privacy. Better methods for image misuse prevention are needed in the future.

Our contributions can be summarized as follows:

- We propose a framework that utilizes generative autoencoders as watermark attackers.
- We show how two instances of generative models, VAEs and diffusions can fit into our framework.
- We evaluate our framework extensively and demonstrate the vulnerabilities of existing watermarks.

2. Related work

Image watermarking and steganography. Steganography and invisible watermarking are key techniques in information hiding, serving diverse purposes such as copyright protection, privacy-preserved communication, and content provenance. Early works in this area employ handcrafted methods, such as Least Significant Bit (LSB) embedding (Wolfgang & Delp, 1996), which subtly hides data in the lowest order bits of each pixel in an image. Over time, numerous techniques have been developed to imperceptibly embed secrets in the spatial (Ghazanfari et al., 2011) and frequency (Holub & Fridrich, 2012; Pevnỳ et al., 2010) domains of an image. Additionally, the emergence of deep learning has contributed significantly to this field. Deep learning methods offer improved robustness against noise while maintaining the quality of the generated image. SteganoGAN (Zhang et al., 2019a) uses generative adversarial networks (GAN) for steganography and perceptual image optimization. RivaGAN (Zhang et al., 2019b), further improves GAN-based watermarking by leveraging attention mechanisms. SSL watermarking (Fernandez et al., 2021), trained with self-supervision, enhances watermark features through data augmentation.

Image denoising. Image denoising is a fundamental yet continuously evolving field in low-level vision, as it plays a crucial role in numerous practical applications. Over the past few decades, several models have been developed to capture image priors for denoising, including nonlocal self-similarity approaches like BM3D (Dabov et al., 2007), which utilizes a two-stage non-locally collaborative filtering method. In recent years, deep neural networks have been applied to address the denoising problem. Zhang et al. (2017) introduced residual learning and batch standardization into image denoising through feed-forward denoising CNNs (DnCNN). Another highly flexible and effective CNN denoiser is DPIR (Zhang et al., 2021), which employs a plug-and-play framework. Image denoising methods can also be applied to remove hidden messages in invisible watermarks (Hosam, 2019).

Deep generative models. The high-dimensional nature of images poses unique challenges to generative modeling. In response to these challenges, several types of deep generative models have been developed, including Variational Auto-Encoders (VAEs) (Vincent et al., 2008; Van Den Oord et al., 2017), Generative Adversarial Networks (GANs) (Goodfellow et al., 2020), Flow-based generative models (Rezende & Mohamed, 2015), and Diffusion models (Ho et al., 2020; Rombach et al., 2022). These models leverage deep latent representations to generate high-quality synthetic images and approximate the true data distribution. In this paper, we aim to demonstrate the capability of these models in removing invisible watermarks from images by utilizing the latent representations obtained through the encoding and decoding processes.

3. Preliminaries

3.1. Problem setup

In this work, we focus on addressing invisible watermarks, as visible watermarks can be handled using existing tools like image inpainting.

We approach watermarking as a post-processing methodology comprised of two algorithms: Watermark and Detect. (1) The Watermark algorithm embeds an invisible watermark mk into an original image x to produce a watermarked image $\hat{x}$, such that $\hat{x}$ looks virtually identical to x. Formally, $\hat{x} \leftarrow \mathsf{Watermark}(x)$. (2) The Detect algorithm extracts the embedded watermark mk from a suspect image. If no watermark is found or the extracted watermark does not match mk, detection fails. Otherwise, it succeeds.

We consider the following threat model for image watermarking:

Adversary's capabilities. We assume that the adversary has black-box input-output access to the watermarking model or only has access to the watermarked images. This adversary is capable of modifying the images using arbitrary side information and computational resources.

Adversary's objective. The primary objective of the adversary is to render the watermark detection algorithm ineffective. Specifically, the adversary aims to produce an image $\tilde{x}$ for which the Detect algorithm fails. Simultaneously, the output image $\tilde{x}$ should also maintain comparable quality to

the original, non-watermarked image.

3.2. Watermarking methods

This paper reviews and evaluates several well-established digital watermarking techniques, which can serve as post-processing methods for embedding watermarks. The assessment covers a range of approaches, from traditional signal processing techniques to state-of-the-art deep learning methods.

DWT-DCT-SVD based watermarking. The DWT-DCT-SVD watermarking method (He & Hu, 2018) combines Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), and Singular Value Decomposition (SVD) to embed watermarks in color images. First, the RGB color space of the cover image is converted to YUV. DWT is then applied to the Y channel, and DCT divides it into blocks. SVD is performed on each block. Finally, the watermark is embedded into the blocks.

RivaGAN watermarking. RivaGAN (Zhang et al., 2019b) presents a robust image watermarking method using GANs. It employs two adversarial networks to assess watermarked image quality and remove watermarks. An encoder embeds the watermark, while a decoder extracts it. By combining these, RivaGAN offers superior performance and robustness.

StegaStamp watermarking. StegaStamp (Tancik et al., 2020) is a CNN method that exhibits exceptional robustness. It uses differentiable image perturbations in training and a spatial transformer network to resist small perspective changes.

SSL watermarking. SSL watermarking (Fernandez et al., 2021) utilizes pre-trained neural networks' latent spaces to encode watermarks. Networks pretrained with self-supervised learning (SSL) extract effective features for watermarking. The method embeds watermarks through back-propagation and data augmentation, then detects and decodes them from the watermarked image or its features.

3.3. Classical attacking methods

In this section, we review attacking methods for invisible watermarks.

Image transformation attack. For image transformations, we modify brightness and contrast. We also test JPEG compression, which compresses images by quantizing rounded discrete cosine transform coefficients of 8x8 blocks.

Image denoising attack. We use BM3D (Dabov et al., 2007) and DPIR (Zhang et al., 2021) as two image denoising methods. BM3D utilizes a two-stage non-locally collaborative filtering method. DPIR is a deep learning-based denoising method.

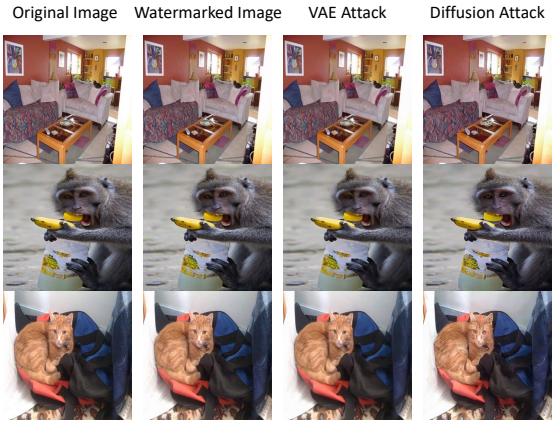

Figure 3. Examples of watermarking attack. The watermark (RivaGAN watermarking) is undetectable in the attacked images. VAE attack tends to make the image blurry. More examples can be found in Figure 4.

4. Watermark removal with generative autoencoders

In this section, we first describe generative autoencoders and our motivation for using them to erase watermarks. We then introduce variational autoencoders and diffusion models as instances we implement.

4.1. Generative autoencoders as watermark erasers

Autoencoders (Kramer, 1991; 1992) learn efficient data encodings by training an encoder E_ϕ that maps data x to a latent space $\mathcal{Z}$ and a decoder D_θ that reconstructs x from $\mathcal{Z}$. Usually, $\mathcal{Z}$ has lower dimension than x, indicating information loss.

We define generative autoencoders to be generative models that can be learned to sample from a true distribution $p(x)$. They are usually learned by minimizing some kind of reconstruction loss that compares the reconstructed $\hat{x} = D_\theta(E_\phi(x))$ with the original data x for a set X of samples from $p(x)$. To use the learned model to sample from $p(x)$, a usual approach is to first take a random sample z from $\mathcal{Z}$ and then decode it, i.e., computing $D_\theta(z)$.

To erase watermarks from a watermarked data point $\hat{x}$, we encode it with $E_\phi(\hat{x})$ and then decode the encoding, i.e., computing $\tilde{x} = D_\theta(E_\phi(\hat{x})) = D_\theta(E_\phi(\mathsf{Watermark}(x)))$.

4.2. Motivation and assumptions

The motivation for using generative autoencoders (VAEs) as invisible watermark erasers is based on two assumptions.

First, we assume that $E_\phi(x)$ and $E_\phi(\mathsf{Watermark}(x))$ are indistinguishable, i.e. the encoder E_ϕ can remove enough information from data such that the encoding of the original

data x cannot be distinguished from that of the watermarked $\hat{x}$. If this assumption holds, the reconstruction watermarked data $D_\theta(E_\phi(x))$ should resemble the reconstruction from original data $D_\theta(E_\phi(\hat{x}))$, i.e., watermark-free.

Second, we assume that the original distribution $p(x)$, from which the model learns to sample, is watermark-free, i.e., we have access to generative autoencoders that are trained on data that do not contain learnable patterns of invisible watermarks. If this assumption holds, when we sample from these models, it's likely that we get watermark-free generations. While we cannot say for sure how many images in the dataset contain invisible watermarks, it's safe to say that even if they do, their invisible watermarks are created with different methods and different messages.

4.3. Variational autoencoders

To learn to sample from $p(x)$, variational autoencoders (Kingma & Welling, 2013) consider the joint distribution $p(x, z)$ of both the data and the latent variable. $\log p(x)$ is optimized using the evidence lower bound (ELBO), i.e.

$$\log p(x) \geq \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p(x,z)}{q_\phi(z|x)} \right], \tag{1}$$

which can further be dissected into two terms, a reconstruction term and a prior matching term

$$\mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z) \right] - D_{\mathrm{KL}}(q_\phi(z|x) \| p(z)). \tag{2}$$

Here $q_\phi(z|x)$ is the encoder distribution and $p_\theta(x|z)$ is the decoder distribution. $p(z)$ is the prior distribution of the latent space z. The encoder is often chosen to be a multivariate Gaussian, while the prior is a standard Gaussian:

$$\begin{aligned} q_\phi(z|x) &= \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x) I) \\ p(z) &= \mathcal{N}(z; 0, I). \end{aligned} \tag{3}$$

4.4. Diffusion models

Diffusion models (Ho et al., 2020) define a generative process that learns to sample from an unknown true distribution $p(x)$. To learn this process, Gaussian noise is added to some original sample $x_0 \sim p(x)$ iteratively from time step 0 through time step T. The distribution of the noise data x_t at time step t is

$$q(x_t|x_0) = \mathcal{N}(\alpha(t)x_0, \sigma^2(t) I), \tag{4}$$

where $\alpha(t), \sigma(t)$ are functions describing the noise schedule.

The denoising process predicts original data x_0 with x_t using a predictor function x_θ.

A diffusion model can be understood as a hierarchical VAE (Luo, 2022; Kingma et al., 2021). To use a diffusion model as a watermark attacker, we pick a small time step $t_0 < T$, and add noise to the data until that time step. This noising process from time step 0 to t_0 is considered the encoder E_ϕ. Denoising from x_t is considered the decoder D_θ.

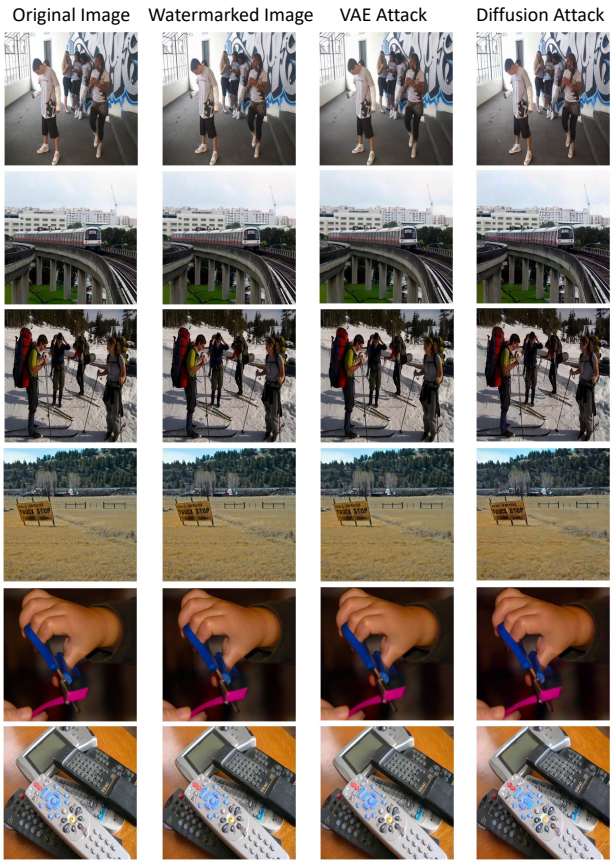

Figure 4. We show more experimental results here including the failure cases. Specifically, we observe that the diffusion attack exhibits limitations when applied to images containing human faces and text. VAE attack tends to over-smooth the image and make it blurry.

5. Experiment

5.1. Experiment setup

Model and datasets. For variational autoencoders, we test two pre-trained image compression models: Bmshj2018 (Ballé et al., 2018) and Cheng2020 (Cheng et al., 2020) from the CompressAI library zoo (Bégaint et al., 2020). For diffusion models, we choose `stable-diffusion-v1-5` from Stable Diffusion (Rombach et al., 2022). The experiments are conducted on a sample of 100 randomly selected images from the MS-COCO dataset (Lin et al., 2014).

Watermark settings. We set the number of bits for watermarking methods as $k = 32$. The tested attacks include brightness change (0.5), contrast change (0.5), and JPEG

Attacker	PSNR↑	SSIM↑	FID↓	Bit Acc↓	Word Acc↓
DCT-DWT-SVD based watermarking:					
Brightness	12.07	0.707	21.16	**0.443**	0.03
Contrast	18.34	0.801	16.99	**0.443**	0.02
JPEG	31.93	0.906	35.00	0.688	0.00
BM3D	33.37	0.896	90.41	0.576	0.00
DPIR	34.84	0.945	18.47	0.918	0.21
Bmshj2018	31.02	0.873	77.11	0.526	**0.00**
Cheng2020	31.96	0.887	69.03	**0.525**	**0.00**
Diffusion	24.88	0.712	41.37	0.643	**0.00**
RivaGAN watermarking:					
Brightness	12.05	0.705	30.87	0.992	0.87
Contrast	18.34	0.802	25.43	0.995	0.89
JPEG	32.05	0.906	35.92	0.959	0.41
BM3D	33.43	0.896	91.59	0.950	0.34
DPIR	34.96	0.945	18.77	0.996	0.87
Bmshj2018	31.07	0.873	78.45	**0.648**	**0.00**
Cheng2020	32.03	0.888	67.82	**0.636**	**0.00**
Diffusion	24.82	0.706	45.25	**0.629**	**0.00**
SSL watermarking:					
Brightness	12.08	0.705	32.38	0.999	0.99
Contrast	18.37	0.803	29.80	1.000	1.00
JPEG	32.05	0.904	43.19	0.805	0.01
BM3D	33.67	0.897	93.13	**0.671**	0.00
DPIR	35.10	0.945	26.94	0.937	0.26
Bmshj2018	31.14	0.874	80.63	**0.640**	**0.00**
Cheng2020	32.10	0.887	71.08	**0.634**	**0.00**
Diffusion	24.83	0.707	47.42	0.719	**0.00**
StegaStamp watermarking:					
Brightness	12.03	0.727	51.72	1.000	1.00
Contrast	17.97	0.805	51.29	1.000	1.00
JPEG	26.89	0.840	72.04	1.000	1.00
BM3D	28.57	0.874	118.14	1.000	1.00
DPIR	27.67	0.876	58.17	1.000	1.00
Bmshj2018	27.75	0.847	93.36	1.000	1.00
Cheng2020	28.33	0.868	85.72	1.000	1.00
Diffusion	22.63	0.622	69.65	**0.648**	**0.50**

Table 1. Performance of attacks on different watermarking methods. For every watermarking method, we **bold** the bit accuracies and word accuracies for the top-3 most successful attacks. For the top-3 most successful attacks, we underline the best image quality metric. Generative autoencoders (with gray background) are usually the most effective in erasing watermarks.

Watermark	PSNR↑	SSIM↑	FID↓	Bit Acc↑	Word Acc↑
DWT-DCT-SVD	39.47	0.983	6.66	1.00	1.00
RivaGAN	40.55	0.979	14.36	1.00	1.00
SSL	41.78	0.985	25.27	1.00	1.00
StegaStamp	28.36	0.909	49.10	1.00	1.00

Table 2. Performance of different watermarking methods. All methods successfully detect the embedded watermark.

compression (quality 50). The denoising methods used are BM3D with a standard deviation of 0.1 and DPIR with a noise level of 5. The compression factors are set to 3 for Bmshj2018 and Cheng2020, and the number of noise steps is set to 20 for diffusion models.

Evaluation metrics. We evaluate the quality of attacked and watermarked images compared to the original cover image using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Frechet Inception Distance (FID) (Heusel et al., 2017). To assess the robustness of the watermark, we measure bit accuracy (the percentage of correctly decoded bits) and word accuracy (the accuracy at the word level).

5.2. Experiment results and analysis

We report the watermark removal results in Table 1 and the quality and detection rate for watermarked images in Table 2. All four watermark methods can successfully embed messages in the image and recover them.

Watermark removal. Generative autoencoders successfully erased all watermarks from the first three methods (DCD-DWT-SVD, RivaGAN, and SSL) at the word accuracy level and consistently performed as the top attacker at the bit accuracy level. StegaStamp watermarking proved to be the most challenging to erase empirically. Only the diffusion model managed to eliminate a significant number of watermarks (50% on average), while other models failed to remove any. However, as shown in Table 2, StegaStamp also exhibited the lowest watermarked image quality, indicating a trade-off between the quality of watermarked images and the robustness of watermark detection.

Image quality reservation. Among the top 3 successful watermark removers, we underline the best quality metric. VAE tends to be better in terms of PSNR and SSIM, while diffusion is better in terms of FID. We manually checked a small batch of images to check their visual quality. As shown in the examples in Figure 3 and Figure 4, VAE-generated images tend to be blurry, which corresponds well to the fact that SSIM and PSNR are known to be unable to measure the blurring of images (Ndajah et al., 2010; Wang et al., 2004). Therefore, we conclude that diffusion models are better in terms of visual quality.

6. Discussion and conclusion

In this paper, we investigate the vulnerabilities and threats to invisible watermarks. Through extensive experiments, we show that malicious adversaries can remove watermarks without damaging the visual quality by leveraging advanced generative autoencoders. We hope that our results can encourage the community to rethink how to protect the copyright of the images in the future.

References

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Bégaint, J., Racapé, F., Feltman, S., and Pushparaja, A. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7939–7948, 2020.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8): 2080–2095, 2007.

Fernandez, P., Sablayrolles, A., Furon, T., J'egou, H., and Douze, M. Watermarking images in self-supervised latent spaces. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3054–3058, 2021.

Ghazanfari, K., Ghaemmaghami, S., and Khosravi, S. R. Lsb++: An improvement to lsb+ steganography. In *TENCON 2011-2011 IEEE Region 10 Conference*, pp. 364–368. IEEE, 2011.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Google. Google keynote (google i/o 23). *Google blog*, 2023. URL https://io.google/2023/program/396cd2d5-9fe1-4725-a3dc-c01bb2e2f38a/.

He, Y. and Hu, Y. A proposed digital image watermarking based on dwt-dct-svd. In *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1214–1218. IEEE, 2018.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Holub, V. and Fridrich, J. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*, pp. 234–239. IEEE, 2012.

Hosam, O. Attacking image watermarking and steganography-a survey. *International Journal of Information Technology and Computer Science*, 11(3):23–37, 2019.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

Kramer, M. A. Autoassociative neural networks. *Computers & chemical engineering*, 16(4):313–328, 1992.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Luo, C. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

Ndajah, P., Kikuchi, H., Yukawa, M., Watanabe, H., and Muramatsu, S. Ssim image quality metric for denoised images. In *Proc. 3rd WSEAS Int. Conf. on Visualization, Imaging and Simulation*, pp. 53–58, 2010.

Pevnỳ, T., Filler, T., and Bas, P. Using high-dimensional image models to perform highly undetectable steganography. In *Information Hiding: 12th International Conference, IH 2010, Calgary, AB, Canada, June 28-30, 2010, Revised Selected Papers 12*, pp. 161–177. Springer, 2010.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

Tancik, M., Mildenhall, B., and Ng, R. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2117–2126, 2020.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Wiggers, K. Microsoft pledges to watermark ai-generated images and videos. *TechCrunch blog*, 2023. URL https://techcrunch.com/2023/05/23/microsoft-pledges-to-watermark-ai-generated-images-and-videos/.

Wolfgang, R. B. and Delp, E. J. A watermark for digital images. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pp. 219–222. IEEE, 1996.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., and Timofte, R. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021.

Zhang, K. A., Cuesta-Infante, A., Xu, L., and Veeramachaneni, K. Steganogan: High capacity image steganography with gans. *arXiv preprint arXiv:1901.03892*, 2019a.

Zhang, K. A., Xu, L., Cuesta-Infante, A., and Veeramachaneni, K. Robust invisible video watermarking with attention. *ArXiv*, abs/1909.01285, 2019b.