Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data (Supplementary Material)

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

1. Implementation Details

Steganography encoder. The encoder is trained to embed a fingerprint into an image while minimizing the pixel difference between the input and stego images. We follow the technical details in [2]. The binary fingerprint vector is first passed through a fully-connected layer and then reshaped as a tensor with one channel dimension and with the same spatial dimension of the cover image. We then concatenate this fingerprint tensor and the image along the channel dimension as the input to a U-Net architecture [1]. The output of the encoder, the stego image, has the same size as that of the input image. Note that passing the fingerprint through a fully-connected layer allows for every bit of the binary sequence to be encoded over the entire spatial dimensions of the input image and flexible to the image size. The fingerprint length is set to 100 as suggested in [2]. The length of 100 bits leads to a large enough space for fingerprint allocation while not having a side effect on the fidelity performance. We visualize an example of encoder architecture in Figure 1 with image size 128×128 for CelebA and LSUN *Bedroom*. For the other image sizes, the architectures are simply scaled up or down with more or fewer layers.

Steganography decoder. The decoder is trained to detect the hidden fingerprint from the stego image. We follow the technical details in [2]. It consists of a series of convolutional layers with kernel size 3x3 and strides ≥ 1, dense layers, and a sigmoid output activation to produce a final output with the same length as the binary fingerprint vector. We visualize an example of decoder architecture in Figure 2 with image size 128×128 for CelebA and LSUN *Bedroom*. For the other image sizes, the architectures are simply scaled up or down with more or fewer layers.

Steganography training. The encoder and decoder are jointly trained end-to-end w.r.t. the objective in Eq. 1 in the main paper and with randomly sampled fingerprints. The encoder is trained to balance fingerprint detection and image reconstruction. At the beginning of training, we set $\lambda = 0$ to focus on fingerprint detection, otherwise, fingerprints cannot be accurately embedded into images. After the fingerprint detection accuracy achieves 95% (that takes 3-5 epochs), we increase λ linearly up to 10 within 3k iterations to shift our focus more on image reconstruction. We train the encoder and decoder for 30 epochs in total. Given the batch size of 64, it takes about 0.5/2/4 hours to jointly train a 32/128/256-resolution encoder and decoder using 1 NVIDIA Tesla V100 GPU with 16GB memory.

2. Additional Samples

See Figure 3, 4, 5, 6, and 7 for fingerprinted samples on a variety of generation applications, models, and datasets. We obtain the same conclusion as in Section 5.3 in the main paper: The fingerprints are imperceptibly transferred to the generative models and then to generated images.

3. Robustness of ProGAN on LSUN *Bedroom*

We in additional experiment on the robustness of ProGAN on LSUN *Bedroom*. We plot the bitwise accuracy w.r.t. the amount of perturbations in Figure 8. We obtain the same conclusions as those in Section 5.4 in the main paper. In specific, the working range w.r.t. each perturbation: Gaussian noise standard deviation $\sim [0.0, 0.1]$, Gaussian blur kernel size $\sim [0, 7]$, JPEG compression quality $\sim [30, 100]$, and center cropping size $\sim [108, 128]$, which are reasonably wide ranges in practice.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1

[*] Equal contribution.

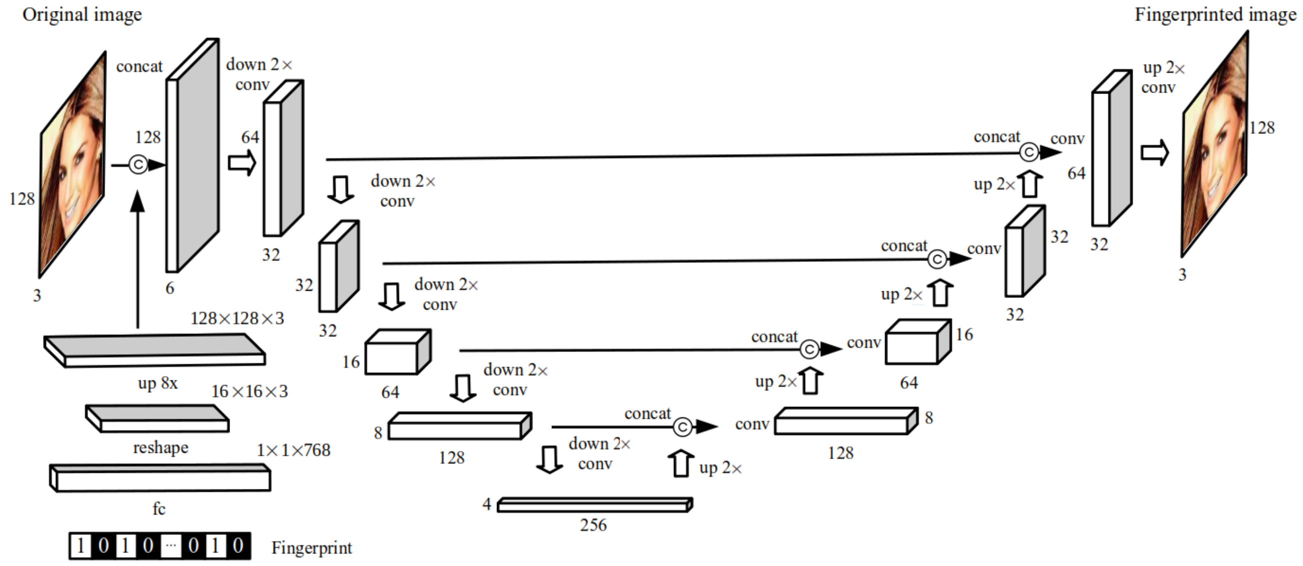


Figure 1: Steganography encoder architecture.

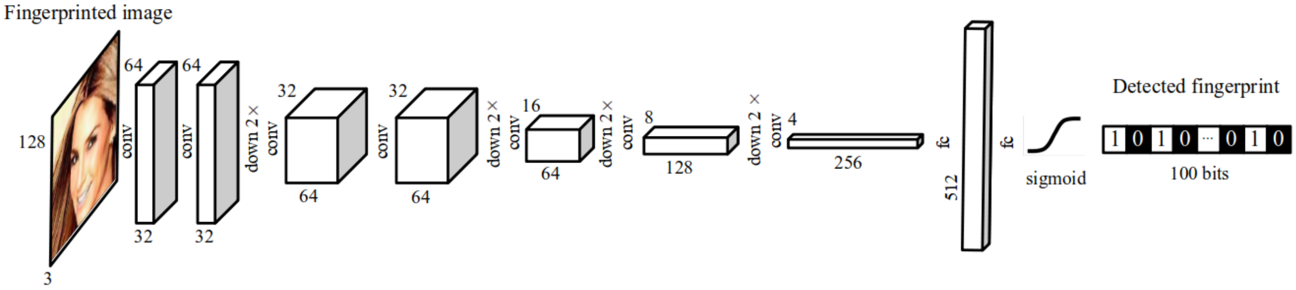
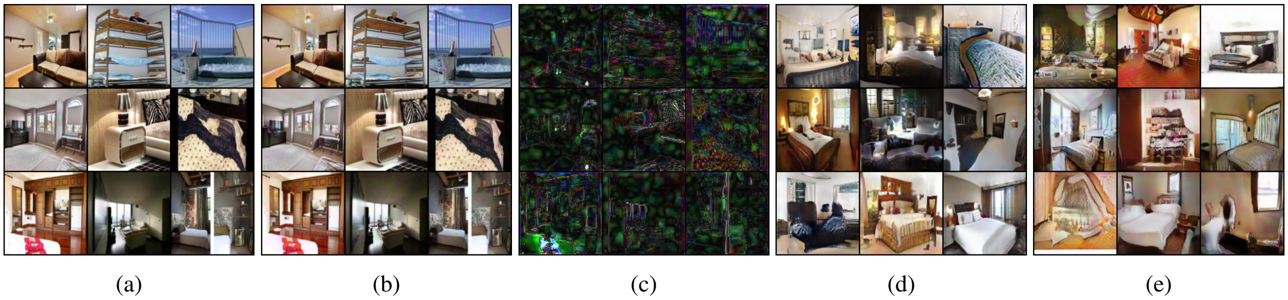


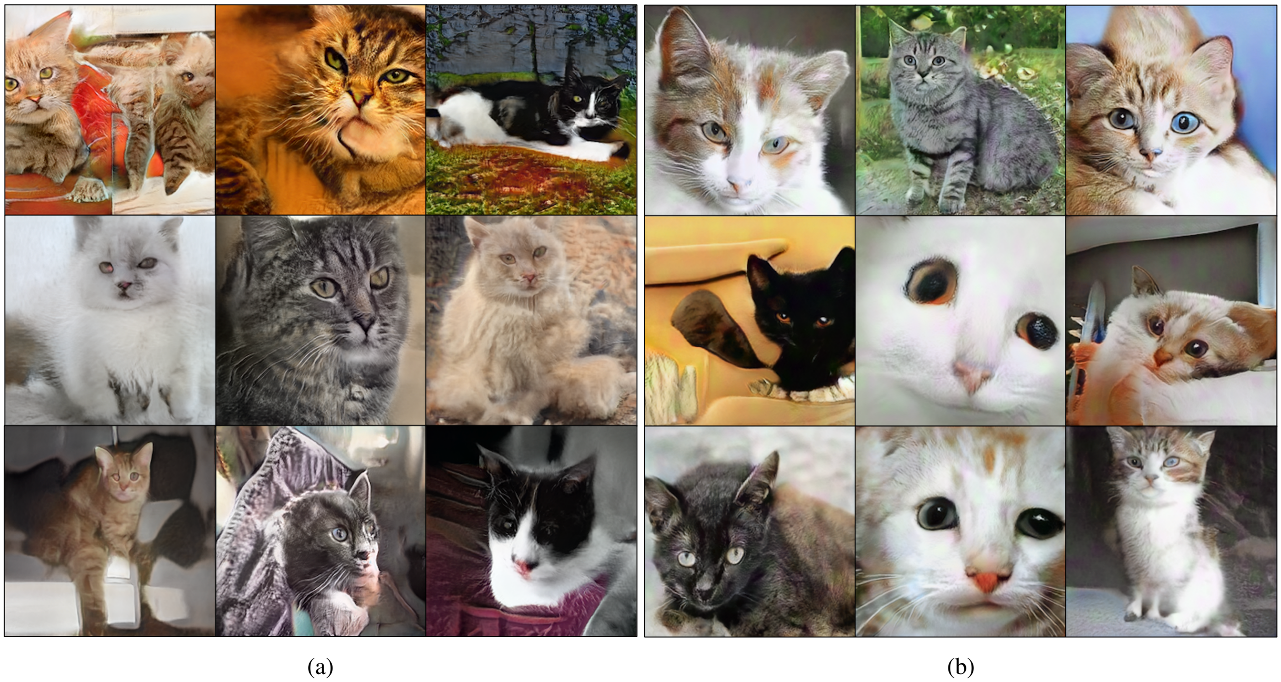
Figure 2: Steganography decoder architecture.

[2] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *CVPR*, 2020. 1



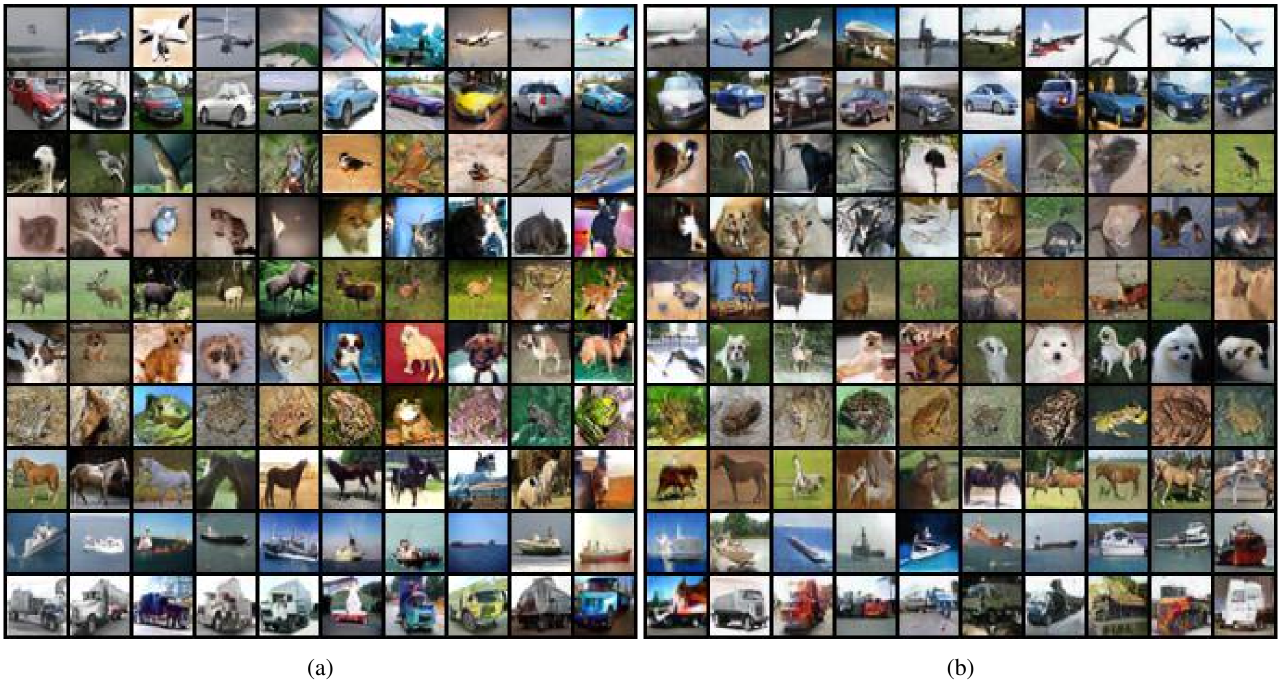
(a) (b) (c) (d) (e)

Figure 3: LSUN *Bedroom* samples at 128×128 for Table 1 last two columns in the main paper, supplementary to Figure 2 in the main paper. (a) Original real training samples. (b) Fingerprinted real training samples. (c) The difference between (a) and (b), 10× magnified for easier visualization. (d) Samples from the non-fingerprinted ProGAN. (e) Samples from the fingerprinted ProGAN.



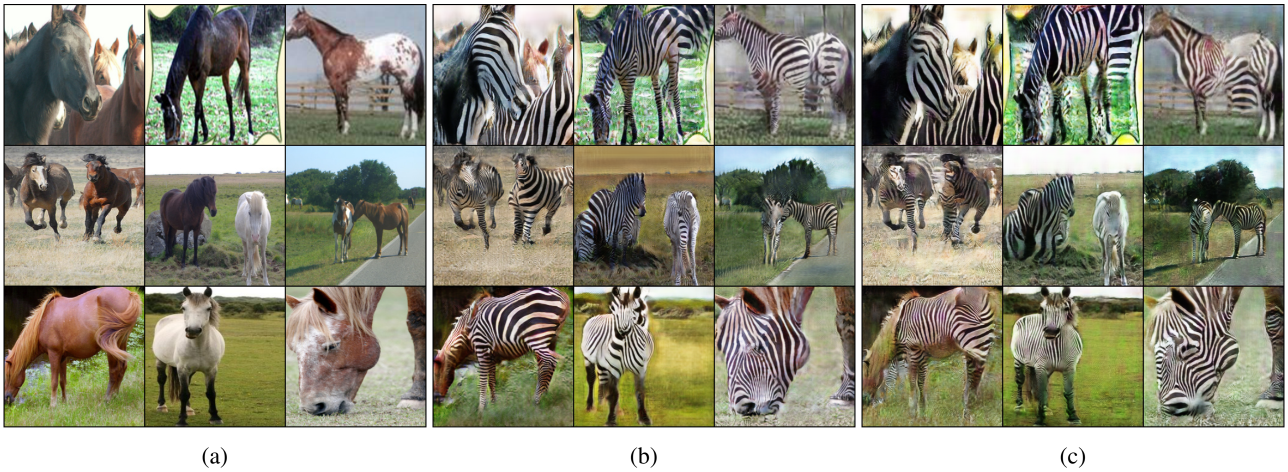
(a) (b)

Figure 4: LSUN *Cat* samples at 256×256 for Table 1 last two columns in the main paper, supplementary to Figure 2 in the main paper. (a) Samples from the non-fingerprinted StyleGAN2. (b) Samples from the fingerprinted StyleGAN2.


(a)

(b)

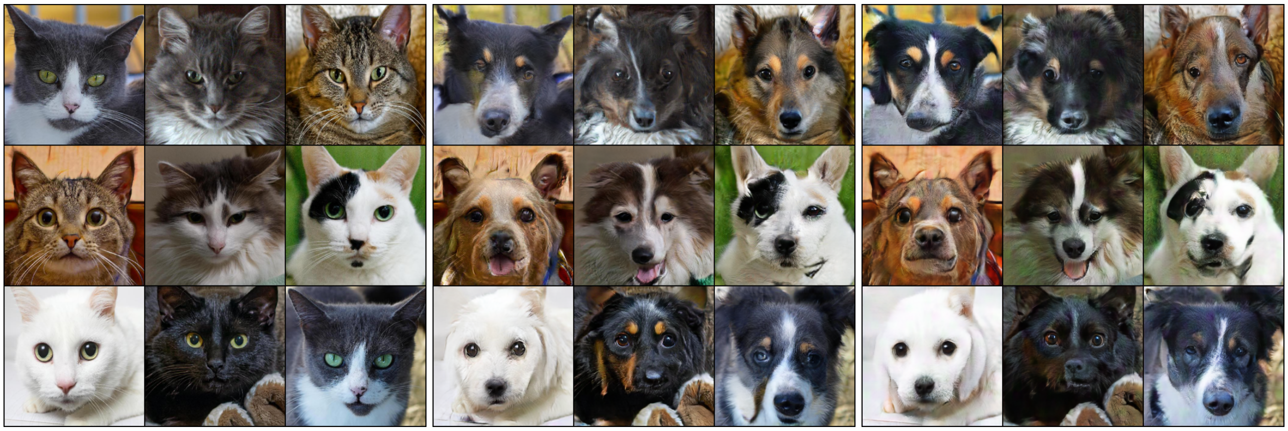
Figure 5: CIFAR-10 samples at 32×32 for Table 1 last two columns in the main paper, supplementary to Figure 2 in the main paper. (a) Samples from the non-fingerprinted BigGAN. (b) Samples from the fingerprinted BigGAN.


(a)

(b)

(c)

Figure 6: *Horse→Zebra* samples at 256×256 for Table 1 last two columns in the main paper, supplementary to Figure 2 in the main paper. (a) Real source samples for input conditioning. (b) Samples from the non-fingerprinted CUT. (c) Samples from the fingerprinted CUT.



(a) (b) (c)

Figure 7: *Cat*→*Dog* samples at 256×256 for Table 1 last two columns in the main paper, supplementary to Figure 2 in the main paper. (a) Real source samples for input conditioning. (b) Samples from the non-fingerprinted CUT. (c) Samples from the fingerprinted CUT.

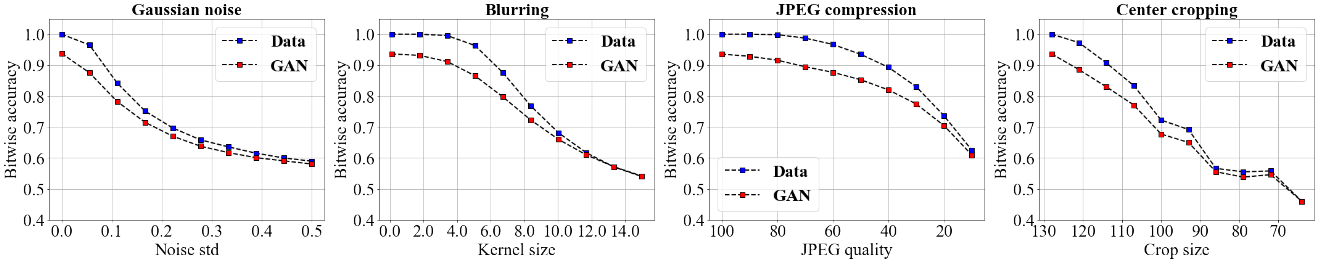


Figure 8: Red plots show the artificial fingerprint detection in bitwise accuracy w.r.t. the amount of perturbations over ProGAN trained on LSUN *Bedroom*. Blue dots represent detection accuracy on the fingerprinted real training images, which serve as the upper bound references for the red dots. This is supplementary to Figure 3 in the main paper.