

Hiding Images in Diffusion Models by Editing Learned Score Functions

Haoyu Chen, Yunqiao Yang, Nan Zhong, and Kede Ma*

City University of Hong Kong

{haoychen3-c, yqyang.cs}@my.cityu.edu.hk, {nzhong, kede.ma}@cityu.edu.hk

<https://github.com/haoychen3/DMIH/>

Abstract

Hiding data using neural networks (i.e., neural steganography) has achieved remarkable success across both discriminative classifiers and generative adversarial networks. However, the potential of data hiding in diffusion models remains relatively unexplored. Current methods exhibit limitations in achieving high extraction accuracy, model fidelity, and hiding efficiency due primarily to the entanglement of the hiding and extraction processes with multiple denoising diffusion steps. To address these, we describe a simple yet effective approach that embeds images at specific timesteps in the reverse diffusion process by editing the learned score functions. Additionally, we introduce a parameter-efficient fine-tuning method that combines gradient-based parameter selection with low-rank adaptation to enhance model fidelity and hiding efficiency. Comprehensive experiments demonstrate that our method extracts high-quality images at human-indistinguishable levels, replicates the original model behaviors at both sample and population levels, and embeds images orders of magnitude faster than prior methods. Besides, our method naturally supports multi-recipient scenarios through independent extraction channels.

1. Introduction

The evolution of data hiding has paralleled advancements in digital media, progressing from traditional bitstream manipulations [8] to deep learning paradigms [1, 3, 60]. A prevailing scheme of *hiding data with neural networks* adopts an autoencoder architecture, where an encoding network embeds a secret message into some cover media, and a decoding network is responsible for retrieving the message. Despite demonstrated feasibility, its practical deployment faces three fundamental constraints. First, the requirement for secure transmission of the decoding network creates logistical vulnerabilities [10, 11]. Second, state-of-the-art steganalysis tools [4, 12, 18, 19, 33, 52, 57] can detect the

embedded message from the cover media (commonly in the forms of digital images and videos) with high accuracy, leading to compromised secrecy. Last, multi-recipient scenarios necessitate complex key design and management [3, 25, 53, 55, 65].

A paradigm shift emerges with *hiding data in neural networks* [5, 32, 44, 50, 51], where secret data is embedded directly into model parameters. This approach evades conventional steganalysis methods, which are typically designed to analyze stego multimedia rather than neural network weights. Initially, such hiding methods showed success in discriminative models [32, 44, 50, 51]. However, recent attention has shifted towards generative models due to their increased utility and wider adoption. Moreover, generative models can directly produce secret data without the need for a separate decoding network, thus inherently resolving transmission security issues. For instance, Chen *et al.* [5] proposed to embed images at specific locations of the learned distributions by deep generative models, which has proven effective with SinGANs [41], a variant of generative adversarial networks (GANs) [14].

The advent of diffusion models [21, 47] introduces new opportunities for this generative probabilistic hiding approach. Despite slightly different purposes (backdoor or watermarking), existing approaches [6, 7, 36] face critical bottlenecks (see Table 1). First, they are inadequate in hiding complex natural images of rich structures and textures, with reconstruction peak signal-to-noise ratio (PSNR) ≤ 25 dB. Second, they compromise model fidelity, with Fréchet inception distance (FID) [20] degradation exceeding 100%, raising detectability concerns. Last, they require full model retraining or fine-tuning (≥ 10 GPU hours) to learn parallel secret diffusion processes.

In this paper, we describe a simple yet effective approach to hiding images in diffusion models. We identify that editing the learned score functions (by inserting secret key-to-image mappings) at specific timesteps enables precise image embedding without disrupting the original chain of the reverse diffusion process. The stego diffusion model is pub-

*Corresponding author.

Table 1. Comparison of diffusion-based data hiding methods in terms of extraction accuracy (high: PSNR > 43.59 dB for 32 × 32 images, ensuring visual imperceptibility), model fidelity (high: FID variations ≤ 20%), hiding efficiency (high: ≤ 0.2 GPU hours for 256 × 256 images), and scalability (support for multi-recipients).

| Method | Extraction Accuracy | Model Fidelity | Hiding Efficiency | Scalability |
|----------------------|---------------------|----------------|-------------------|-------------|
| StableSignature [11] | High | Low | Low | No |
| AquaLoRA [10] | High | Low | High | No |
| BadDiffusion [7] | Low | Low | Low | No |
| TrojDiff [6] | High | Low | Low | No |
| WDM [36] | Low | Low | Low | No |
| Ours | High | High | High | Yes |

licly shared¹, replicating the original model behaviors in synthesizing in-distribution high-quality images. The secret image extraction is achieved in a single step via key-guided and timestep-conditioned “denoising.” To further improve model fidelity and hiding efficiency, we introduce a hybrid parameter-efficient fine-tuning (PEFT) method that combines gradient-based parameter selection [15] and low-rank adaptation (LoRA) [23], which reduces trainable parameters by 86.3% compared to full fine-tuning. Comprehensive experiments demonstrate the effectiveness of our method across four critical dimensions: 1) extraction accuracy, achieving 52.90 dB in PSNR for 32 × 32 images and 39.33 dB for 256 × 256 images; 2) model fidelity, maintaining nearly original FID scores (4.77 versus 4.79 on CIFAR10 [28]) with minimal sample-level distortion; 3) hiding efficiency, reducing embedding time to 0.04 and 0.18 GPU hours for 32 × 32 and 256 × 256 images, respectively; and 4) scalability, simultaneously embedding four images for different recipients with independent extraction keys.

2. Related Work

Neural Steganography. We focus primarily on hiding data in neural networks [5, 32, 44, 50, 51], which involves embedding secret data within neural network parameters or structures, ensuring covert communication without compromising model performance. Representative neural steganography strategies encompass replacing least significant bits of model parameters [32, 44], substituting redundant parameters [32, 51], mapping parameter values [32, 44, 50] or signs [32, 44] directly to secret messages, memorizing secret-labeled synthetic data [44], and hiding images in deep probabilistic models [5]. Despite their effectiveness, diffusion models have not been extensively considered in neural steganography.

Diffusion Models represent a significant advancement in generative modeling, offering high-quality data synthesis through multi-step denoising diffusion processes. Inspired by non-equilibrium thermodynamics [43] and score match-

ing in empirical Bayes [24, 38], diffusion models define a forward diffusion process, which progressively adds noise to data until it becomes random, and a reverse (generative) process, which learns to iteratively remove this noise, reconstructing the original data. In contrast to GANs, diffusion models optimize a likelihood-based objective [21, 46], typically resulting in more stable training and better mode coverage. Architecturally, diffusion models predominantly employ U-Nets [40] and Transformers [48], operating in either pixel [21, 27, 47] or latent [35, 37, 39] space.

Existing methods for hiding data in diffusion models typically involve concurrent training of secret reverse diffusion processes [6, 7, 36]. As a result, these approaches suffer from limited extraction accuracy, noticeable degradation in model fidelity, and high computational demands, especially when applied to complex natural images. Diffusion model editing techniques [13, 29, 54, 56], originally designed to alter or remove learned concepts and rules, may offer a potential remedy. Some have been adapted to embed invisible yet recoverable signals for watermarking purposes [10, 11]. Nevertheless, these editing methods do not fulfill the precise image reconstruction requirements we are looking for.

Parameter-Efficient Fine-Tuning. PEFT methods can be broadly classified into two categories: selective and reparameterized fine-tuning. Selective methods strategically identify only a subset of parameters for adjustment based on simple heuristics [59] or parameter gradients [17, 63]. In contrast, reparameterized fine-tuning introduces additional trainable parameters to a pre-trained frozen backbone. Notable examples include prompt-tuning [30], adapters [22], and LoRA [23] and its variants [16, 26, 34, 61]. This paper presents a hybrid PEFT method for hiding images in pixel-space diffusion models, effectively combining the strengths of both selective and reparameterized methods.

3. Hiding Images in Diffusion Models

In this section, we first provide the necessary preliminaries of diffusion models, which serve as the basis of the proposed method. Subsequently, we describe the typical image hiding scenario, and present in detail our diffusion-based

¹Only the secret key of few bits is shared between the sender and recipient via a secure subliminal channel.

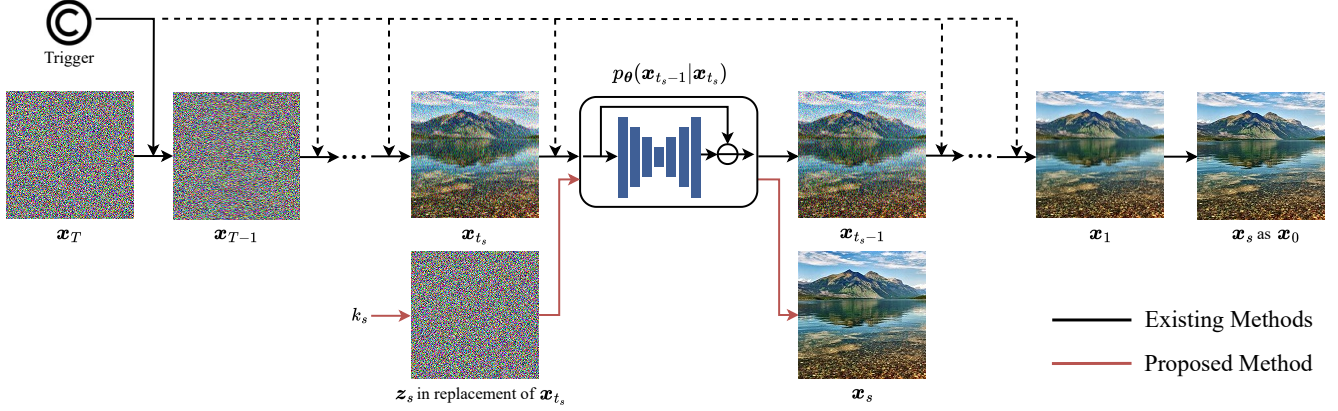


Figure 1. Comparison of diffusion-based image hiding methods. Existing methods typically embed trigger patterns (acting as secret keys) at the initial timestep of the reverse diffusion process, and optionally at all subsequent timesteps (denoted by dashed lines). These patterns guide the reconstruction of the secret image \mathbf{x}_s , but compromise model fidelity and reduce hiding efficiency due to persistent intervention of the entire reverse diffusion process. In stark contrast, the proposed method operates selectively: the secret image \mathbf{x}_s is embedded and extracted only at a privately chosen timestep t_s . Hiding is governed by a secret key k_s , which serves as the seed to generate the input Gaussian noise \mathbf{z}_s . By localizing the intervention to a single timestep, the integrity of the reverse diffusion process is preserved.

steganography method (see Fig. 1).

3.1. Preliminaries

Diffusion models, notably denoising diffusion probabilistic models (DDPMs) [21], have recently emerged as state-of-the-art deep generative models capable of synthesizing high-quality images through a defined diffusion process.

Forward Diffusion Process. Starting from a clean image $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is iteratively added over T timesteps following a variance schedule $\{\beta_1, \dots, \beta_T\}$, described as

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (1)$$

where each step is governed by

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right). \quad (2)$$

Eventually, this process transforms the clean image \mathbf{x}_0 into nearly pure Gaussian noise \mathbf{x}_T .

Reverse Diffusion Process. A Markov chain with parameters θ is learned to iteratively recover the clean image from noise, modeled as

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \quad (3)$$

Model parameters θ are optimized by minimizing a denoising objective [21]:

$$\min_{\theta} \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon, t)\|_2^2 \right], \quad (4)$$

where $\epsilon_\theta(\cdot, \cdot)$ is the learned noise estimation function (also known as the score function), implemented typically using a U-Net, and $\bar{\alpha}_t = \prod_{i=1}^t (1 - \beta_i)$ controls the noise scale at timestep t . Eq. (4) can also be viewed as a variational bound of the log-likelihood of the observed data \mathbf{x}_0 .

Inference iteratively employs the trained and timestep-conditioned score function with optimal parameters $\bar{\theta}$ to transform Gaussian noise into a clean image:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_{\bar{\theta}}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}, \quad (5)$$

for $t \in \{T, \dots, 1\}$, where $\alpha_t = 1 - \beta_t$, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and σ_t controls stochasticity at timestep t , with $\sigma_t^2 = (1 - \bar{\alpha}_{t-1})\beta_t / (1 - \bar{\alpha}_t)$. σ_1 is usually set to zero to remove randomness at the final timestep. For simplicity, we slightly abuse the notation \mathbf{x}_t to denote both the noisy image and its denoised counterpart at timestep t .

3.2. Image Hiding Scenario

We consider a single-image hiding scenario following the formulation of the classical prisoner’s problem by Simmons [42], which involves three distinct parties:

- A sender, who embeds a secret image \mathbf{x}_s in a diffusion model with a secret key \mathcal{K}_s , and subsequently shares the stego diffusion model publicly;
- A recipient, who utilizes the privately shared secret key \mathcal{K}_s to extract the secret image \mathbf{x}_s from the publicly transmitted stego diffusion model;
- An inspector, who examines the publicly available diffusion models to verify that they retain normal generation functionality, and do not exhibit any suspicious behavior.

The key objectives in this scenario are twofold: ensuring high accuracy in the extraction of the secret image by the authorized recipient, and preserving the sample-level and population-level fidelity of the diffusion model in image generation so that it remains undetected during scrutiny by the inspector.

3.3. Proposed Method

Basic Idea. Existing diffusion-based techniques [6, 7, 36] often embed secret images by modifying the entire Markov chain of the reverse diffusion process, resulting in significant challenges regarding extraction accuracy, model fidelity, and hiding efficiency. To resolve these, we propose a simple yet effective hiding method that operates at a single timestep. This process uses a secret key $\mathcal{K}_s = \{k_s, t_s\}$, where k_s is the seed to deterministically generate Gaussian noise z_s , and t_s is the selected timestep for image hiding. The secret key \mathcal{K}_s is privately communicated with the recipient, enabling precise, secure, and rapid one-step reconstruction of the secret image \mathbf{x}_s as

$$\mathbf{f}_{\tilde{\theta}}(z_s, t_s) = \frac{1}{\sqrt{\bar{\alpha}_{t_s}}} \left(z_s - \sqrt{1 - \bar{\alpha}_{t_s}} \epsilon_{\tilde{\theta}}(z_s, t_s) \right), \quad (6)$$

where $\tilde{\theta}$ denotes the optimal parameters after editing.

Loss Function. To ensure minimal distortion in reconstructing the secret image \mathbf{x}_s , we define the extraction accuracy loss as

$$\ell_a(\theta; z_s, t_s, \mathbf{x}_s) = \|\mathbf{f}_{\theta}(z_s, t_s) - \mathbf{x}_s\|_2^2. \quad (7)$$

By optimizing the learned score function $\epsilon_{\theta}(z_s, t_s)$ using the accuracy loss, the secret image is embedded within the diffusion model. Meanwhile, it is crucial that the stego diffusion model closely replicates the generative functionality of the original model. Therefore, we introduce an additional model fidelity loss to regulate the edited score function:

$$\ell_f(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\|\epsilon_{\theta}(\mathbf{x}_t, t) - \epsilon_{\tilde{\theta}}(\mathbf{x}_t, t)\|_2^2 \right], \quad (8)$$

where $\epsilon_{\tilde{\theta}}(\cdot, \cdot)$ denotes the original score function and $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$. In contrast to the loss $\ell_a(\theta)$ in Eq. (7), which penalizes only the reconstruction error of the secret image \mathbf{x}_s at the specific timestep t_s , the model fidelity loss $\ell_f(\theta)$ computes the expectation of the residual between noise estimated by the original and edited score functions. This expectation is taken across uniformly sampled timesteps $t \sim \text{Uniform}(\{1, \dots, T\})$, clean images $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, and Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to ensure complete adherence to the original diffusion process. Moreover, the loss $\ell_f(\theta)$ eliminates the need to retrain the original diffusion model or access the original training dataset, requirements typically imposed by existing methods [6, 7, 36]. Lastly, the overall loss function is a linear weighted summation of the two terms:

$$\ell(\theta; z_s, t_s, \mathbf{x}_s) = \ell_a(\theta; z_s, t_s, \mathbf{x}_s) + \lambda \ell_f(\theta), \quad (9)$$

where λ is a trade-off parameter.

Multi-image Hiding. We extend the single-image hiding method to scenarios involving multiple recipients, each with a unique secret key. For a set of secret images $\mathcal{X}_s = \{\mathbf{x}_s^{(1)}, \dots, \mathbf{x}_s^{(M)}\}$, where $\mathbf{x}_s^{(i)}$ denotes the secret image intended for the i -th recipient, and M is the total number of secret images, equal to the number of recipients. Each secret image $\mathbf{x}_s^{(i)}$ is paired with a secret key $\mathcal{K}_s^{(i)} = \{k_s^{(i)}, t_s^{(i)}\}$, forming the key collection $\mathcal{K}_s = \{\mathcal{K}_s^{(1)}, \dots, \mathcal{K}_s^{(M)}\}$ distributed among M different recipients. The loss function in Eq. (9) is then modified as

$$\ell(\theta; \mathcal{K}_s, \mathcal{X}_s) = \frac{1}{M} \sum_{i=1}^M \ell_a(\theta; z_s^{(i)}, t_s^{(i)}, \mathbf{x}_s^{(i)}) + \lambda \ell_f(\theta), \quad (10)$$

where $z_s^{(i)}$ is the Gaussian noise generated using the seed $k_s^{(i)}$. After optimizing the pre-trained diffusion model using $\ell(\theta; \mathcal{K}_s, \mathcal{X}_s)$, the i -th recipient can extract her/his secret image $\mathbf{x}_s^{(i)}$ using the designated secret key $\mathcal{K}_s^{(i)}$. Without access to additional secret keys, each recipient is prevented from retrieving secret images intended for others.

3.4. Score Function Editing by PEFT

To improve both model fidelity and hiding efficiency, we introduce a hybrid PEFT technique that combines the advantages of selective and reparameterized PEFT strategies. Our method consists of three steps: 1) computing parameter-level sensitivity with respect to the editing loss function (in Eq. (9)), 2) identifying the most sensitive layers, and 3) applying reparameterized fine-tuning to these selected layers.

Parameter Sensitivity Calculation. We quantify the importance of each parameter through gradient-based sensitivity analysis [17]. For parameter θ_i in the score function $\epsilon_{\theta}(z_s, t_s)$, the task-specific sensitivity g_i is approximated by accumulating squared gradients over N iterations:

$$g_i = \sum_{j=1}^N \left(\frac{\partial \ell(\theta)}{\partial \theta_i^{(j)}} \right)^2, \quad (11)$$

where i and j are the parameter and iteration indices, respectively. This metric identifies parameters most responsive to the hiding objective.

Sensitive Layer Selection. We extend parameter-level to layer-level sensitivity through spatial aggregation. Specifically, we begin by binarizing the sensitivity score of each parameter g_i :

$$b_i = \begin{cases} 1 & g_i \geq \tau \\ 0 & g_i < \tau \end{cases}, \quad (12)$$

where τ is a predefined threshold that controls the sensitive parameter sparsity. Next, we rank all network layers by their counts of sensitive parameters (*i.e.*, those with

Table 2. Extraction accuracy comparison for 32×32 and 256×256 secret images. “ \uparrow ”: larger is better. The top-2 results are highlighted in boldface.

| Method | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | DISTS \downarrow |
|------------------------------------|-----------------|-----------------|--------------------|--------------------|
| 32×32 | | | | |
| Baluja17 [3] | 25.40 | 0.89 | 0.116 | 0.051 |
| HiDDeN [65] | 25.24 | 0.88 | 0.252 | 0.075 |
| Weng19 [53] | 26.66 | 0.93 | 0.059 | 0.035 |
| HiNet [25] | 30.39 | 0.94 | 0.033 | 0.026 |
| PRIS [55] | 29.83 | 0.94 | 0.041 | 0.027 |
| Chen22 [5] | 47.72 | 0.99 | 0.001 | 0.002 |
| BadDiffusion [7] | 22.08 | 0.86 | 0.129 | 0.060 |
| TrojDiff [6] | 46.54 | 0.99 | 0.001 | 0.004 |
| WDM [36] | 36.49 | 0.99 | 0.003 | 0.008 |
| Ours | 52.90 | 0.99 | 0.001 | 0.001 |
| 256×256 | | | | |
| Baluja17 [3] | 26.46 | 0.90 | 0.200 | 0.089 |
| HiDDeN [65] | 27.13 | 0.91 | 0.233 | 0.100 |
| Weng19 [53] | 33.85 | 0.95 | 0.089 | 0.047 |
| HiNet [25] | 35.31 | 0.96 | 0.087 | 0.041 |
| PRIS [55] | 37.42 | 0.97 | 0.050 | 0.029 |
| Chen22 [5] | 36.44 | 0.96 | 0.073 | 0.035 |
| BadDiffusion [7] | 17.68 | 0.81 | 0.386 | 0.137 |
| TrojDiff [6] | 24.74 | 0.94 | 0.057 | 0.076 |
| WDM [36] | 17.97 | 0.83 | 0.245 | 0.144 |
| Ours | 39.33 | 0.97 | 0.043 | 0.018 |

Table 3. Model fidelity and hiding efficiency comparison for 32×32 and 256×256 secret images. Embedding time is measured in terms of GPU hours per image.

| Method | FID \downarrow | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | DISTS \downarrow | Time \downarrow |
|------------------------------------|------------------|-----------------|-----------------|--------------------|--------------------|-------------------|
| 32×32 | | | | | | |
| Original | 4.79 | N/A | N/A | N/A | N/A | N/A |
| BadDiffusion | 6.88 | 23.78 | 0.80 | 0.222 | 0.082 | 4.87 |
| TrojDiff | 4.64 | 28.72 | 0.91 | 0.114 | 0.049 | 12.72 |
| WDM | 5.09 | 22.50 | 0.84 | 0.228 | 0.083 | 2.35 |
| Ours | 4.77 | 31.06 | 0.94 | 0.077 | 0.037 | 0.04 |
| 256×256 | | | | | | |
| Original | 7.46 | N/A | N/A | N/A | N/A | N/A |
| BadDiffusion | 15.75 | 16.40 | 0.60 | 0.452 | 0.224 | 31.92 |
| TrojDiff | 14.36 | 18.73 | 0.70 | 0.407 | 0.169 | 83.68 |
| WDM | 15.07 | 18.59 | 0.67 | 0.445 | 0.200 | 19.22 |
| Ours | 8.39 | 23.31 | 0.83 | 0.235 | 0.112 | 0.18 |

$b_i = 1$), and select top- η layers for subsequent structured and computationally efficient fine-tuning. Critically, we avoid summing raw sensitivity scores within layers, as this could bias selection towards layers with disproportionately many parameters—even if those parameters are less impactful (known as the law of triviality). By prioritizing the number of sensitive parameters over their cumulative sensitivity, we ensure a balanced layer-wise assessment.

LoRA-based Fine-tuning. For each selected sensitive layer, we implement a variant of LoRA for PEFT. Specifi-

cally, for linear layers, we apply the standard LoRA [23] by injecting trainable low-rank matrices $\Delta\mathbf{W} = \mathbf{A}\mathbf{B}$, where $\mathbf{A} \in \mathbb{R}^{m \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times n}$ with rank $r \ll \min\{m, n\}$. For convolutional layers, we first reshape the 4D convolution filters into 2D matrices by flattening the input channel and filter dimensions, and then apply LoRA accordingly. To improve fine-tuning performance and accelerate convergence, we incorporate the rank stabilization trick proposed in [26], adjusting the scaling factor to be proportional to $O(1/\sqrt{r})$. Meanwhile, we apply the learning rate decoupling trick introduced by [16], setting different learning rates for the matrices \mathbf{A} and \mathbf{B} , respectively.

4. Experiments

In this section, we first describe the experimental setups. Subsequently, we present comprehensive comparisons against nine state-of-the-art neural steganography methods across three key dimensions: extraction accuracy, model fidelity, and hiding efficiency. Finally, we extend our analysis to multi-image hiding scenarios, demonstrating the scalability of our approach.

4.1. Experimental Setups

Implementation Details. We employ the pre-trained DDPMs [21] as our default pixel-space diffusion models at two image resolutions: 32×32 pixels using CIFAR10 [28] and 256×256 pixels using LSUN bedroom [58]. Unlike previous approaches [6, 7, 36], which typically embed overly simplistic image patterns like QR codes or icons, we assess our method with complex natural images aggregated from widely used datasets, including COCO [31], DIV2K [2], LSUN Church [58], and Places [64].

The architectures and hyperparameters of the DDPMs follow the specifications outlined in [21]. By default, secret embedding and extraction are performed at timestep $t_s = 500$. For parameter sensitivity analysis, we accumulate gradients over $N = 50$ iterations. The threshold in Eq. (12) is selected to achieve sensitive parameter sparsity levels of 0.01 for 32×32 images and 0.1 for 256×256 images. Correspondingly, the number of sensitive layers η is set to 15 for 32×32 images and 45 for 256×256 images, respectively. The maximum iteration number for LoRA is set to 2,000, with rank $r = 64$ for 32×32 images and $r = 128$ for 256×256 images.

Evaluation Criteria. Our evaluation encompasses three aspects: 1) extraction accuracy, quantified via PSNR, the structural similarity (SSIM) index [49], the learned perceptual image patch similarity (LPIPS) measure [62], and the deep image structure and texture similarity (DISTS) metric [9] between the original and extracted secret images; 2) model fidelity, assessed at both the sample-level using PSNR, SSIM, LPIPS and DISTS, and the population-level using FID [20] to measure deviations between generated

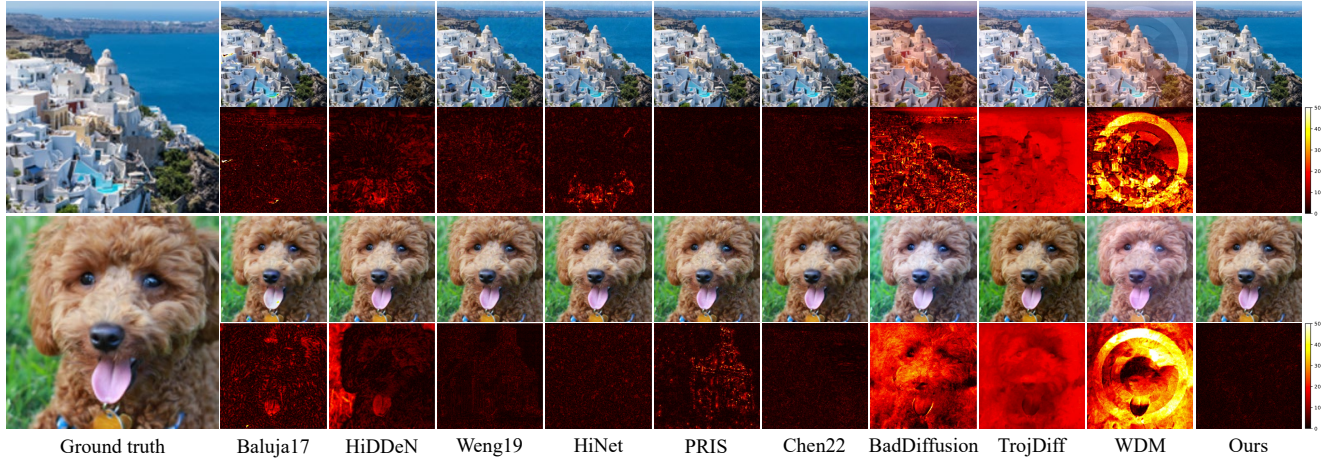


Figure 2. Visual comparison of extracted secret images along with the absolute error maps.



Figure 3. Visual comparison of images generated by the original and stego diffusion models with the same set of initial noise.

images from the original and stego diffusion models; and 3) hiding efficiency, measured by GPU hours required for embedding a secret image on an AMD EPYC 7F52 16-Core CPU and NVIDIA GeForce RTX3090 GPUs.

4.2. Main Results

We compare our method with five autoencoder-based algorithms: 1) Baluja17 [3], 2) HiDDeN [65], 3) Weng19 [53], 4) HiNet [25] and 5) PRIS [55], one GAN-based technique: 6) Chen22 [5], and three diffusion-based methods: 7) BadDiffusion [7], 8) TrojDiff [6] and 9) WDM [36].

Extraction Accuracy. As shown in Table 2, autoencoder-based techniques [3, 25, 53, 55, 65] achieve higher extraction accuracy at higher image resolutions. In contrast, existing GAN-based [5] and diffusion-based [6, 7, 36] methods perform better at lower image resolutions. Moreover, our approach outperforms all existing neural steganography methods across both image resolutions. Fig. 2 presents a qualitative comparison of extraction accuracy using absolute error maps between the original and extracted secret images at the resolution of 256×256 . The visual results indicate that our method produces the least reconstruction errors across all spatial locations.

Model Fidelity. Table 3 shows that our method maintains the closest model fidelity to the original DDPMs, as evidenced by minimal FID variations (4.77 versus 4.79 for 32×32 images and 8.39 versus 7.46 for 256×256 images). Sample-level fidelity metrics (*i.e.*, PSNR, SSIM, LPIPS, and DISTS) also validate the ability of our method to produce visually indistinguishable outputs from the original model. Fig. 3 compares the 256×256 images generated by the original and different stego diffusion models, using the same set of initial noise. It is clear that our method produces images that closely resemble those from the original model, whereas other stego diffusion models introduce noticeable discrepancies in structural details and color appearances of objects and backgrounds.

Hiding Efficiency. As also reported in Table 3, our method significantly reduces the hiding time to only 0.04 GPU hours for 32×32 images and 0.18 GPU hours for 256×256 images, making it over 50 times faster than the second best method, WDM [36]. This substantial efficiency improvement arises because we embed secret images only at a privately chosen timestep t_s , rather than fine-tuning the entire reverse diffusion process over numerous iterations.

Hiding Multiple Images. To explore scalability, we further

Table 4. Extraction accuracy and model fidelity of our method when hiding multiple secret images for different recipients at two image resolutions.

| Measure | # | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | DISTS \downarrow |
|------------------------------------|---|-----------------|-----------------|--------------------|--------------------|
| 32 \times 32 | | | | | |
| Extraction Accuracy | 1 | 52.90 | 0.99 | 0.001 | 0.001 |
| | 2 | 50.44 | 0.99 | 0.001 | 0.001 |
| | 4 | 49.38 | 0.99 | 0.001 | 0.001 |
| Model Fidelity | 1 | 31.06 | 0.94 | 0.077 | 0.037 |
| | 2 | 30.86 | 0.95 | 0.067 | 0.035 |
| | 4 | 30.93 | 0.95 | 0.064 | 0.033 |
| 256 \times 256 | | | | | |
| Extraction Accuracy | 1 | 39.33 | 0.97 | 0.043 | 0.018 |
| | 2 | 38.61 | 0.97 | 0.034 | 0.023 |
| | 4 | 38.31 | 0.96 | 0.058 | 0.029 |
| Model Fidelity | 1 | 23.31 | 0.83 | 0.235 | 0.112 |
| | 2 | 20.53 | 0.78 | 0.313 | 0.137 |
| | 4 | 17.78 | 0.74 | 0.394 | 0.165 |

Table 5. Ablation on the number of iterations for parameter sensitivity accumulation.

| # of Iterations | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | DISTS \downarrow |
|----------------------------|-----------------|-----------------|--------------------|--------------------|
| <i>Extraction Accuracy</i> | | | | |
| 1 | 40.93 | 0.99 | 0.001 | 0.006 |
| 50 | 52.90 | 0.99 | 0.001 | 0.001 |
| 100 | 53.08 | 0.99 | 0.001 | 0.001 |
| <i>Model Fidelity</i> | | | | |
| 1 | 28.58 | 0.92 | 0.101 | 0.046 |
| 50 | 31.06 | 0.94 | 0.077 | 0.037 |
| 100 | 31.20 | 0.94 | 0.074 | 0.036 |

Table 6. Ablation on the number of selected sensitive layers.

| # of Sensitive Layers | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | DISTS \downarrow |
|----------------------------|-----------------|-----------------|--------------------|--------------------|
| <i>Extraction Accuracy</i> | | | | |
| 5 | 47.48 | 0.99 | 0.001 | 0.002 |
| 15 | 52.90 | 0.99 | 0.001 | 0.001 |
| 45 | 54.04 | 0.99 | 0.001 | 0.001 |
| <i>Model Fidelity</i> | | | | |
| 5 | 31.55 | 0.95 | 0.068 | 0.035 |
| 15 | 31.06 | 0.94 | 0.077 | 0.037 |
| 45 | 27.87 | 0.91 | 0.122 | 0.048 |

assess our method in scenarios involving multiple secret images intended for different recipients. Table 4 demonstrates that although embedding additional images slightly sacrifices model fidelity, our method maintains high extraction accuracy, supporting secure communication among multiple recipients.

In summary, our comprehensive experiments confirm that the proposed method achieves superior extraction accuracy, high model fidelity, and exceptional hiding efficiency, establishing a new benchmark in diffusion-based neural steganography.

4.3. Ablation Studies

To systematically analyze the contributions of key components in the proposed method, we conduct comprehensive ablation studies on 32×32 images.

Parameter Sensitivity Accumulation. We first evaluate how the number of iterations N , used for sensitivity accumulation in Eq. (11) impacts performance. As detailed in Table 5, we observe significant performance gains when increasing N from 1 to 50 in terms of both extraction accuracy and model fidelity. However, further increasing N to 100 yields negligible benefits relative to the computational overhead incurred. Consequently, we set $N = 50$ as a reasonable trade-off between method performance and computational efficiency.

Number of Selected Layers. We next analyze how the selection of different numbers of sensitive layers affects performance. As demonstrated in Table 6, selecting too few layers (*e.g.*, 5) results in inadequate extraction accuracy (PSNR of 47.48), despite good model fidelity. Conversely, selecting too many layers (*e.g.*, 45) reduces model fidelity due to excessive parameter adjustments. A balanced selection of 15 layers significantly enhances extraction accuracy (PSNR of 52.90) without compromising model fidelity, and thus is adopted as the default choice.

Secret Timestep Selection. The proposed method introduces flexibility in the choice of the secret embedding timestep t_s . Fig. 4 illustrates how varying the choice of t_s affects performance. Encouragingly, extraction accuracy and model fidelity remain consistently robust across a broad range of timesteps, underscoring the practical advantage and reliability of our approach in various operational (*e.g.*, multi-image and multi-recipient) scenarios.

Number of PEFT Iterations. We further explore how varying the number of iterations in our PEFT method influences performance. As shown in Table 7, increasing the number of PEFT iterations generally improves extraction accuracy but reduces model fidelity and substantially increases computational cost, as expected. For example, with fewer iterations (*e.g.*, 1,000), our method yields high model fidelity but suffers from inadequate extraction accuracy. Thus, we choose 2,000 iterations as our default configuration, as it achieves a balanced trade-off between extraction accuracy, model fidelity, and hiding efficiency.

PEFT versus Full Fine-tuning. We compare our PEFT strategy against full fine-tuning to evaluate its effectiveness and efficiency. The results in Table 8 clearly demonstrate that our PEFT method significantly outperforms full fine-tuning in maintaining model fidelity. Notably, this superior fidelity is achieved with minimal sacrifice in extraction accuracy and a significant reduction in computational time—requiring only half the time of full fine-tuning.

Robustness to Image Noise. To assess robustness against potential input noise, we intentionally perturb secret nat-

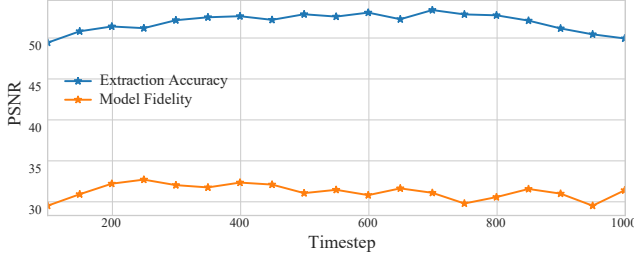


Figure 4. Ablation on the selection of the secret timestep.

Table 7. Ablation on the number of PEFT iterations.

| # of Iterations | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | DISTS \downarrow | Time \downarrow |
|----------------------------|-----------------|-----------------|--------------------|--------------------|-------------------|
| <i>Extraction Accuracy</i> | | | | | |
| 1,000 | 49.31 | 0.99 | 0.001 | 0.001 | 0.02 |
| 2,000 | 52.90 | 0.99 | 0.001 | 0.001 | 0.04 |
| 5,000 | 55.16 | 0.99 | 0.001 | 0.001 | 0.11 |
| <i>Model Fidelity</i> | | | | | |
| 1,000 | 32.54 | 0.96 | 0.067 | 0.032 | – |
| 2,000 | 31.06 | 0.94 | 0.077 | 0.037 | – |
| 5,000 | 25.88 | 0.91 | 0.137 | 0.057 | – |

ural images with random Gaussian noise (with mean zero and variance 100) before embedding. Table 8 confirms the resilience of our method, showing minimal performance degradation under substantial noise conditions. This verifies the potential applicability of our method in encrypted transmission environments.

Compatibility Across Diffusion Model Variants. Lastly, we validate the adaptability of our method by testing it across several popular diffusion model variants, including DDPM [21], DDPM++ [47], DDIM [45], and EDM [27]. Table 9 demonstrates that high extraction accuracy and model fidelity are maintained across these diverse diffusion model architectures, underscoring the broad generalizability and versatility of our method. Note that during DDIM inference, certain denoising steps may be skipped, and this skipping might inadvertently include the selected secret timestep, which explains the best model fidelity.

Overall, these ablation studies collectively validate the efficiency, robustness, and adaptability of our method, firmly establishing its practical effectiveness in neural steganography applications.

5. Conclusion and Discussion

We have proposed a computational method for hiding images in diffusion models by editing learned score functions at specific timesteps, through a hybrid PEFT strategy. Our method achieves outstanding extraction accuracy, preserves model fidelity, and significantly enhances hiding efficiency compared to existing methods. Through extensive experiments and rigorous ablation studies, we highlighted criti-

Table 8. Ablations on PEFT against full fine-tuning, and on hiding clean and noisy natural images.

| Method | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | DISTS \downarrow | Time \downarrow |
|----------------------------|-----------------|-----------------|--------------------|--------------------|-------------------|
| <i>Extraction Accuracy</i> | | | | | |
| Full Fine-tuning | 53.55 | 0.99 | 0.001 | 0.001 | 0.08 |
| Hiding Noisy Image | 51.86 | 0.99 | 0.001 | 0.001 | 0.04 |
| Ours | 52.90 | 0.99 | 0.001 | 0.001 | 0.04 |
| <i>Model Fidelity</i> | | | | | |
| Full Fine-tuning | 23.98 | 0.84 | 0.212 | 0.075 | – |
| Hiding Noisy Image | 31.30 | 0.94 | 0.072 | 0.035 | – |
| Ours | 31.06 | 0.94 | 0.077 | 0.037 | – |

Table 9. Ablation on different diffusion models. Unlike DDPM, DDPM++, and DDIM, the EDM variant is pre-trained on the larger-sized ImageNet dataset (64×64), explaining its slightly lower extraction accuracy and reduced model fidelity.

| Diffusion Model | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | DISTS \downarrow |
|----------------------------|-----------------|-----------------|--------------------|--------------------|
| <i>Extraction Accuracy</i> | | | | |
| DDPM [21] | 52.90 | 0.99 | 0.001 | 0.001 |
| DDPM++ [47] | 53.05 | 0.99 | 0.001 | 0.001 |
| DDIM [45] | 52.96 | 0.99 | 0.001 | 0.001 |
| EDM [27] | 51.30 | 0.99 | 0.002 | 0.001 |
| <i>Model Fidelity</i> | | | | |
| DDPM [21] | 31.06 | 0.94 | 0.077 | 0.037 |
| DDPM++ [47] | 31.37 | 0.95 | 0.073 | 0.035 |
| DDIM [45] | 35.08 | 0.97 | 0.034 | 0.022 |
| EDM [27] | 28.80 | 0.94 | 0.090 | 0.056 |

cal factors contributing to the superior performance of our method. The localized score function editing turns out to be a notable contribution, enabling precise adjustments with minimal computational cost.

Looking ahead, several promising future research directions emerge from this study. One particularly compelling avenue is the development of adaptive hiding strategies that dynamically adjust embedding parameters based on data characteristics or security requirements. Moreover, rigorous theoretical analysis of diffusion model behaviors under various embedding scenarios, including theoretical upper bounds on detectability, will be critical to fully exploit and understand the potential of diffusion-based neural steganography. Lastly, improving embedding robustness against model perturbations—such as noising, pruning, and compression—could transform our method into a highly effective digital watermarking solution, facilitating authentication and copyright protection of diffusion models.

Acknowledgments

This work was supported in part by the Hong Kong RGC General Research Fund (11220224) and the Hong Kong ITC Innovation and Technology Fund (9440390).

References

- [1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *USENIX Security Symposium*, pages 1615–1631, 2018. 1
- [2] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2017. 5
- [3] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems*, pages 2070–2080, 2017. 1, 5, 6
- [4] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5): 1181–1193, 2018. 1
- [5] Haoyu Chen, Linqi Song, Zhenxing Qian, Xinpeng Zhang, and Kede Ma. Hiding images in deep probabilistic models. In *Advances in Neural Information Processing Systems*, pages 36776–36788, 2022. 1, 2, 5, 6
- [6] Weixin Chen, Dawn Song, and Bo Li. TrojDiff: Trojan attacks on diffusion models with diverse targets. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4035–4044, 2023. 1, 2, 4, 5, 6
- [7] Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4015–4024, 2023. 1, 2, 4, 5, 6
- [8] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2007. 1
- [9] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2567–2581, 2020. 5
- [10] Weitao Feng, Wenbo Zhou, Jiyan He, Jie Zhang, Tianyi Wei, Guanlin Li, Tianwei Zhang, Weiming Zhang, and Nenghai Yu. AquaLoRA: Toward white-box protection for customized Stable Diffusion models via watermark LoRA. In *International Conference on Machine Learning*, pages 13423–13444, 2024. 1, 2
- [11] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The Stable Signature: Rooting watermarks in latent diffusion models. In *IEEE/CVF International Conference on Computer Vision*, pages 22409–22420, 2023. 1, 2
- [12] Tong Fu, Liquan Chen, Yinghua Jiang, Ju Jia, and Zhangjie Fu. Image steganalysis based on dual-path enhancement and fractal downsampling. *IEEE Transactions on Information Forensics and Security*, 20:1–16, 2025. 1
- [13] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *IEEE/CVF International Conference on Computer Vision*, pages 2426–2436, 2023. 2
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1
- [15] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024. 2
- [16] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *International Conference on Machine Learning*, pages 17783–17806, 2024. 2, 5
- [17] Haoyu He, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang. Sensitivity-aware visual parameter-efficient fine-tuning. In *IEEE/CVF International Conference on Computer Vision*, pages 11825–11835, 2023. 2, 4
- [18] Jian He, Shaowei Weng, Lifang Yu, Chunyu Zhang, and Wei Chen. An image steganalyzer with comprehensive detection performance. *IEEE Signal Processing Letters*, 30: 1682–1686, 2023. 1
- [19] Jian He, Shaowei Weng, Lifang Yu, and Dewang Chen. Steganalysis network with two-branch preprocessing for spatial and JPEG domains. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(2):1451–1463, 2024. 1
- [20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6627–6638, 2017. 1, 5
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020. 1, 2, 3, 5, 8
- [22] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799, 2019. 2
- [23] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2, 5
- [24] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. 2
- [25] Junpeng Jing, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan. HiNet: Deep image hiding by invertible network. In *IEEE/CVF International Conference on Computer Vision*, pages 4733–4742, 2021. 1, 5, 6
- [26] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023. 2, 5
- [27] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, pages 26565–26577, 2022. 2, 8
- [28] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009. 2, 5

- [29] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision*, pages 22691–22702, 2023. 2
- [30] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021. 2
- [31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and Lawrence C. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014. 5
- [32] Tao Liu, Zihao Liu, Qi Liu, Wujie Wen, Wenyao Xu, and Ming Li. StegoNet: Turn deep neural network into a stego-malware. In *Annual Computer Security Applications Conference*, pages 928–938, 2020. 1, 2
- [33] Ge Luo, Ping Wei, Shuwen Zhu, Xinpeng Zhang, Zhenxing Qian, and Sheng Li. Image steganalysis with convolutional vision Transformer. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3089–3093, 2022. 1
- [34] Hyeon-Woo Nam, Ye-Bin Moon, and Oh Tae-Hyun. Fed-Para: Low-rank Hadamard product for communication-efficient federated learning. In *International Conference on Learning Representations*, 2022. 2
- [35] William Peebles and Saining Xie. Scalable diffusion models with Transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2
- [36] Sen Peng, Yufei Chen, Cong Wang, and Xiaohua Jia. Intellectual property protection of diffusion models via the watermark diffusion process. In *International Conference on Web Information Systems Engineering*, pages 290–305, 2024. 1, 2, 4, 5, 6
- [37] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024. 2
- [38] Martin Raphan and Eero P. Simoncelli. Least squares estimation without priors or supervision. *Neural Computation*, 23(2):374–420, 2011. 2
- [39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2
- [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 2
- [41] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *IEEE/CVF International Conference on Computer Vision*, pages 4570–4580, 2019. 1
- [42] Gustavus J. Simmons. The prisoners’ problem and the subliminal channel. In *Advances in Cryptology*, pages 51–67, 1984. 3
- [43] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015. 2
- [44] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601, 2017. 1, 2
- [45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 8
- [46] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, pages 1415–1428, 2021. 2
- [47] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 2, 8
- [48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5999–6009, 2017. 2
- [49] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5
- [50] Zichi Wang, Guorui Feng, Hanzhou Wu, and Xinpeng Zhang. Data hiding in neural networks for multiple receivers. *IEEE Computational Intelligence Magazine*, 16(4):70–84, 2021. 1, 2
- [51] Zhi Wang, Chaoge Liu, and Xiang Cui. EvilModel: Hiding malware inside of neural network models. In *IEEE Symposium on Computers and Communications*, pages 1–7, 2021. 1, 2
- [52] Kangkang Wei, Weiqi Luo, and Jiwu Huang. Color image steganalysis based on pixel difference convolution and enhanced Transformer with selective pooling. *IEEE Transactions on Information Forensics and Security*, 19:9970–9983, 2024. 1
- [53] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. High-capacity convolutional video steganography with temporal residual modeling. In *International Conference on Multimedia Retrieval*, pages 87–95, 2019. 1, 5, 6
- [54] Tianwei Xiong, Yue Wu, Enze Xie, Zhenguo Li, and Xihui Liu. Editing massive concepts in text-to-image diffusion models. *arXiv preprint arXiv:2403.13807*, 2024. 2
- [55] Hang Yang, Yitian Xu, Xuhua Liu, and Xiaodong Ma. PRIS: Practical robust invertible network for image steganography. *Engineering Applications of Artificial Intelligence*, 133:108419–108427, 2024. 1, 5, 6
- [56] Yunqiao Yang, Long-Kai Huang, Shengzhuang Chen, Kede Ma, and Ying Wei. Learning where to edit vision Transform-

- ers. In *Advances in Neural Information Processing Systems*, pages 134459–134487, 2024. 2
- [57] Weike You, Hong Zhang, and Xianfeng Zhao. A siamese CNN for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 16:291–306, 2020. 1
- [58] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5
- [59] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bit-Fit: Simple parameter-efficient fine-tuning for Transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 1–9, 2022. 2
- [60] Jie Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. Model watermarking for image processing networks. In *AAAI Conference on Artificial Intelligence*, pages 12805–12812, 2020. 1
- [61] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023. 2
- [62] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 5
- [63] Zhi Zhang, Qizhe Zhang, Zijun Gao, Renrui Zhang, Ekaterina Shutova, Shiji Zhou, and Shanghang Zhang. Gradient-based parameter selection for efficient fine-tuning. In *IEEE/CVF International Conference on Computer Vision*, pages 28566–28577, 2024. 2
- [64] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 5
- [65] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. HiDDeN: Hiding data with deep networks. In *European Conference on Computer Vision*, pages 657–672, 2018. 1, 5, 6