

GENERATIVE IMAGE STEGANOGRAPHY WITH EXACT INVERSION AND PROBABILISTIC SHAPING IN LATENT DIFFUSION MODELS

WANTING SUN AND ZHE-MING LU*

School of Aeronautics and Astronautics
Zhejiang University

No. 38, Zheda Road, Hangzhou 310027, P. R. China

wantingsun@zju.edu.cn; *Corresponding author: zheminglu@zju.edu.cn

Received January 2026; revised May 2026

ABSTRACT. *Traditional image steganography embeds secret messages by modifying a given cover image and is therefore vulnerable to increasingly powerful steganalysis. Recent generative steganography based on latent diffusion models offers a promising alternative, but reliable message recovery remains challenging when latent prior compatibility and inversion fidelity must be satisfied simultaneously. In particular, directly mapping discrete bit sequences to continuous latent variables may violate the Gaussian prior of latent diffusion models, while multi-step inversion may accumulate numerical errors and degrade recovery accuracy under lossy perturbations. To address these challenges, we develop a recovery-oriented generative image steganography framework for latent diffusion models. The framework integrates three coordinated components: probabilistic-shaping-guided bit-to-latent mapping for Gaussian-prior compatibility, exact inversion for DPM-Solver++(2M) to suppress cumulative numerical deviation during latent recovery, and interval decoding for robust symbol determination from recovered latent variables. In this way, message embedding and extraction are jointly organized within a unified framework. In addition, two symbol cardinalities, i.e., $M = 2$ and $M = 4$, are investigated to characterize the trade-off between embedding capacity and recovery robustness. Extensive experiments on Stable Diffusion v1.5 and v2.1 show that the proposed method achieves robust message extraction under common distortions while maintaining favorable visual quality and low steganalytic detectability. The results further indicate that binary mapping is more robust to perturbations, whereas 4-ary mapping offers higher payload at the cost of reduced recovery stability.*

Keywords: Image steganography, Generative steganography, Latent diffusion model, Exact inversion, Probabilistic shaping

1. Introduction. Image steganography aims to conceal secret information within seemingly innocuous images so as to avoid arousing suspicion during transmission [1]. Traditional steganographic methods usually embed messages by directly modifying a given cover image. Although this paradigm has been extensively studied, the resulting stego images may still exhibit detectable statistical traces and therefore remain vulnerable to increasingly powerful steganalysis techniques [2, 3]. To alleviate the detectability introduced by direct cover modification, generative steganography has emerged as an alternative paradigm in which the carrier is synthesized together with the hidden message, thereby avoiding explicit cover-alteration traces [1, 3].

The development of generative steganography has closely followed the progress of deep generative models. Early attempts were mainly built upon generative adversarial networks

(GANs) [4]. Some methods embed secret information into class labels or conditional inputs [5, 6], but their payload is inherently limited by the conditioning interface. Other methods inject messages into latent noise variables to synthesize stego images [7, 8, 9]. While such strategies improve generation flexibility, they may compromise image quality and often rely on separately trained extraction modules, which can weaken recovery reliability. Flow-based generative steganography further provides a theoretically invertible route for information recovery [10, 11]. For example, secret messages can be encoded into latent variables and transformed into stego images through an invertible generator, after which recovery is achieved via inverse transformation [10]. However, such methods are still constrained by the generation quality and diversity of the underlying flow-based models.

Recently, diffusion models [12] have become a major framework for high-fidelity image synthesis and have also stimulated growing interest in diffusion-based generative steganography. Representative studies along this line have evolved from DDPM-based schemes, such as StegaDDPM [13], to latent-diffusion-based generative steganography. Within the latter category, different methods emphasize different aspects of the problem. GRDH [14] establishes a robust mapping-based framework on Stable Diffusion and DPM-Solver++, enabling zero-shot text-driven stego image generation and improving robustness under lossy transmission. More recently, StegaMGM [15] explores prompt-conditioned latent-diffusion generative steganography based on text-to-image multimodal generative models, where secret messages are mapped into the initial standard normal latent distribution and recovered through diffusion inversion, leading to improved extraction accuracy under common storage distortions such as JPEG compression.

Despite these advances, reliable message recovery in latent-diffusion-based generative steganography remains insufficiently addressed when two requirements must be satisfied simultaneously: *latent prior compatibility* and *inversion fidelity*. On the one hand, latent diffusion models assume a standard Gaussian prior, whereas directly mapping discrete bit sequences into continuous latent variables may introduce statistical mismatch and thus degrade both generation quality and recovery stability. On the other hand, reverse recovery in diffusion-based generative steganography is inherently multi-step. For high-order samplers such as DPM-Solver++(2M) [16], numerical deviations may accumulate along the inversion trajectory and eventually impair the recoverability of the embedded message.

These observations suggest that reliable message recovery should be understood not as an isolated decoding problem, but as the result of a coordinated design spanning both message embedding and latent recovery. Motivated by this perspective, we develop a recovery-oriented generative image steganography framework for latent diffusion models. The proposed framework organizes three tightly coupled components into a unified chain: probabilistic-shaping-guided bit-to-latent mapping for Gaussian-prior compatibility, exact inversion for controlling cumulative numerical deviation during DPM-Solver++(2M)-based recovery, and interval decoding for robust symbol determination from recovered latent variables. Under this formulation, message embedding and extraction are treated as two coordinated stages of the same latent transmission process. In addition, by considering both $M = 2$ and $M = 4$, the proposed framework enables a systematic analysis of the trade-off between embedding capacity and recovery robustness.

The main contributions of this paper are summarized as follows.

- We develop a recovery-oriented generative image steganography framework for latent diffusion models that jointly considers bit embedding, latent recovery, and message extraction.

- We couple probabilistic-shaping-guided bit-to-latent mapping with exact inversion for DPM-Solver++(2M) in a unified recovery-oriented framework, thereby jointly addressing Gaussian-prior compatibility and inversion stability during message recovery.
- We introduce interval decoding for symbol recovery and systematically investigate two symbol cardinalities, i.e., $M = 2$ and $M = 4$, in order to characterize the trade-off between embedding capacity and recovery robustness.

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries of exact inversion for DPM-Solver++(2M) and the notation adopted in this work. Section 3 presents the proposed framework, including bit mapping, probabilistic shaping, exact inversion, and interval decoding. Section 4 reports the experimental setup and evaluation results, including datasets, baselines, evaluation metrics, implementation details and comparative experiments. Finally, Section 5 concludes the paper and discusses the limitations of the current study as well as possible future directions.

2. Preliminaries on Diffusion Inversion. In this section, we briefly summarize the inversion background and notation used in this work. Our purpose is to establish the numerical setting for latent recovery in diffusion-based generative steganography, rather than to re-derive the inversion algorithm itself. Since the proposed framework performs message extraction by recovering a latent representation from the received stego image, inversion fidelity directly affects the reliability of subsequent symbol recovery.

2.1. Forward sampling formulation and notation. Let z_t denote the latent state at diffusion time t , and let $x_\theta(z_t, t)$ denote the data-prediction network used by the sampler. For latent diffusion models, all sampling and inversion operations are conducted in the latent space. Following the common formulation of diffusion ODE solvers, let σ_t denote the noise schedule, α_t the signal-preservation coefficient, and

$$\lambda_t = \log \frac{\alpha_t}{\sigma_t} \quad (1)$$

the log signal-to-noise ratio (log-SNR). Given a decreasing time grid $\{t_i\}_{i=0}^T$, define

$$h_i = \lambda_{t_i} - \lambda_{t_{i-1}}, \quad i = 1, \dots, T. \quad (2)$$

For DPM-Solver++(2M), the forward update is a second-order multistep approximation, which improves sampling efficiency but also makes the corresponding inversion more sensitive to accumulated numerical deviation.

2.2. Naive DDIM inversion and its limitation. To motivate the use of exact inversion, we first consider the commonly used naive DDIM inversion. For a first-order solver, the forward update can be written as

$$z_{t_i} = \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} z_{t_{i-1}} - \alpha_{t_i} (e^{-h_i} - 1) x_\theta(z_{t_{i-1}}, t_{i-1}). \quad (3)$$

A straightforward reverse approximation is to estimate $z_{t_{i-1}}$ from z_{t_i} by replacing the unknown implicit term with the prediction at the current state, i.e.,

$$\tilde{z}_{t_{i-1}} = \frac{\sigma_{t_{i-1}}}{\sigma_{t_i}} (z_{t_i} + \alpha_{t_i} (e^{-h_i} - 1) x_\theta(z_{t_i}, t_{i-1})). \quad (4)$$

This approximation is simple and computationally efficient, but its accuracy is limited. In particular, when the forward sampler is a high-order multistep solver such as DPM-Solver++(2M), the inversion process is no longer determined only by the current state. Higher-order terms implicitly depend on previous states, and the resulting approximation error may accumulate over multiple inversion steps. In generative steganography,

such errors are especially undesirable because even small latent deviations may lead to incorrect symbol decisions during message extraction.

2.3. Exact inversion background for DPM-Solver++(2M). To obtain a more accurate latent recovery, we adopt the exact-inversion formulation for DPM-Solver++(2M) as the numerical inversion backbone of the proposed framework. The basic idea is to replace the purely explicit reverse approximation with an implicit correction based on the backward Euler viewpoint, together with a high-order term approximation.

For each macro-interval $[t_{i-1}, t_i]$, the interval is first split into J micro-steps, and a coarse reverse trajectory is estimated. Let $\hat{z}_{t_{i-(j-1)/J}}$ denote the current estimate. For the micro-step partition within each macro-interval, $h_{i-j/J}$ denotes the log-SNR increment between two consecutive micro-step time points. The coarse inversion proceeds as

$$\begin{aligned} \bar{z}_{t_{i-j/J}} &= -\alpha_{t_{i-j/J}} (e^{-h_{i-j/J}} - 1) x_\theta(\hat{z}_{t_{i-(j-1)/J}}, t_{i-j/J}), \\ \hat{z}_{t_{i-j/J}} &\leftarrow \frac{\sigma_{t_{i-j/J}}}{\sigma_{t_{i-(j-1)/J}}} \hat{z}_{t_{i-(j-1)/J}} - \bar{z}_{t_{i-j/J}}. \end{aligned} \quad (5)$$

To further reduce the inversion error, a second-order correction is introduced. Let

$$r_i = \frac{h_{i-1}}{h_i}, \quad (6)$$

and define the corrected prediction term as

$$d'_i = x_\theta(\hat{z}_{t_{i-1}}, t_{i-1}) + \frac{1}{2} r_i (x_\theta(\hat{z}_{t_{i-1}}, t_{i-1}) - x_\theta(\hat{z}_{t_{i-2}}, t_{i-2})). \quad (7)$$

Then the backward-Euler-based update is written as

$$z'_{t_i} = \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} \hat{z}_{t_{i-1}} - \alpha_{t_i} (e^{-h_i} - 1) d'_i. \quad (8)$$

Finally, the latent estimate is refined through fixed-point implicit updates,

$$\hat{z}_{t_i} \leftarrow \text{UPDATE}(\hat{z}_{t_{i-1}}, z'_{t_i}; t_{i-1}, t_i), \quad (9)$$

until convergence. In implementation, UPDATE is instantiated as a forward-step-based implicit refinement operator.

The final step from t_1 to t_0 is handled in the same spirit. First, an initial estimate of $\hat{z}_{t_0}$ is formed as

$$\hat{z}_{t_0} = \frac{\sigma_{t_0}}{\sigma_{t_1}} (\hat{z}_{t_1} + \alpha_{t_1} (e^{-h_1} - 1) x_\theta(\hat{z}_{t_1}, t_1)), \quad (10)$$

and then refined via an implicit correction analogous to the previous macro-steps.

In the proposed framework, this inversion background serves only as the numerical basis for latent recovery from the received stego image. The next section describes how this recovery module is integrated with prior-compatible message mapping and interval decoding in the overall framework.

3. Methodology. As illustrated in Figure 1, the proposed method is formulated as a recovery-oriented framework that links secret bitstreams, prior-compatible latent representations, and stego images. Rather than treating latent inversion as an isolated post-processing step, the framework jointly organizes bit mapping, probabilistic shaping, exact inversion-based latent recovery, and interval decoding toward reliable message extraction under perturbation. In the hiding (Tx) stage, the bitstream is grouped into k -bit symbols and probabilistically shaped to match the standard Gaussian prior of the LDM latent; the shaped latent is then advanced by the ODE sampler and decoded by D to produce a stego image. In the extraction (Rx) stage, a possibly degraded stego image is encoded

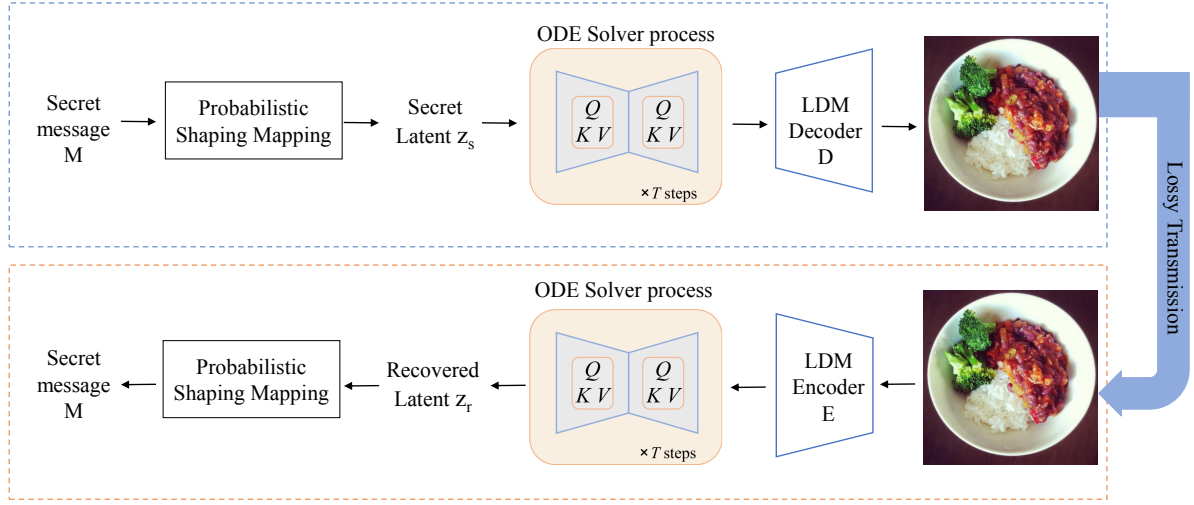


FIGURE 1. Architecture of the proposed system. Probabilistic shaping aligns the latent representation with the LDM prior, while exact inversion and interval decoding are jointly used for message recovery from the received stego image.

Algorithm 1. Tx: Probabilistic Shaping to $\mathcal{N}(0,1)$

Require: Bitstream $\mathbf{b}$ (length B), $M = 2^k$, clipping ε

- 1: $L_{\text{pad}} \leftarrow (k - (B \bmod k)) \bmod k$; pad $\mathbf{b}$; reshape to (N, k) where $N = (B + L_{\text{pad}})/k$
- 2: $s_n \leftarrow \text{bin2dec}$ of each row
- 3: **for** $n = 1$ **to** N **do**
- 4: Draw $u_n \sim \mathcal{U}[0, 1)$ and set $p_n \leftarrow (s_n + u_n)/M$
- 5: $\tilde{p}_n \leftarrow \text{clip}(p_n, \varepsilon, 1 - \varepsilon)$; $z_n \leftarrow \Phi^{-1}(\tilde{p}_n)$
- 6: **end for**
- 7: Write $\mathbf{z}$ into the selected latent positions of the LDM tensor; run the forward sampler $\mathcal{A}$ and decoder D

Ensure: Stego image and recorded padding length L_{pad}

Algorithm 2. Rx: Interval Decoding and Bit Recovery

Require: Received stego image, $M = 2^k$, clipping ε , recorded L_{pad}

- 1: Encode the received stego image by E to obtain an initial latent estimate
- 2: Refine the latent initialization by decoder inversion
- 3: Apply exact inversion $\mathcal{A}^{-1}$ to recover the latent estimate $\hat{\mathbf{z}}$
- 4: **for** $n = 1$ **to** N **do**
- 5: $t_n \leftarrow \text{clip}(\Phi(\hat{z}_n), \varepsilon, 1 - \varepsilon)$
- 6: $\hat{s}_n \leftarrow \min\{M - 1, \max\{0, \lfloor M t_n \rfloor\}\}$
- 7: **end for**
- 8: Convert $\{\hat{s}_n\}$ to bits row-wise and drop the last L_{pad} bits

Ensure: Recovered bitstream $\hat{\mathbf{b}}$

by E and inverted to recover the message-carrying latent representation, from which the original bitstream is reconstructed by interval decoding after removing padding bits. The Tx/Rx procedures are summarized in Algorithm 1 and Algorithm 2.

3.1. Bit mapping and symbol generation. Before latent shaping, the input bitstream is first converted into discrete symbols so that the payload can be represented in a form compatible with interval-based latent mapping and recovery. Let $\mathbf{b} \in \{0, 1\}^B$ denote the input bitstream of length B , and let $M = 2^k$ be the modulation order, where each symbol carries k bits. The required tail padding and symbol count are

$$\begin{aligned} L_{\text{pad}} &= (k - (B \bmod k)) \bmod k, \\ N &= \frac{B + L_{\text{pad}}}{k}, \end{aligned} \quad (11)$$

where L_{pad} is the number of padding bits appended to make the bitstream divisible by k , and N is the resulting number of k -bit symbols. After padding, $\mathbf{b}$ is reshaped into an (N, k) matrix, and each row is converted into a symbol index

$$s_n = \text{bin2dec}(b_{n,1}, \dots, b_{n,k}) \in \{0, \dots, M-1\}. \quad (12)$$

The resulting symbol sequence $\{s_n\}_{n=1}^N$ serves as the discrete input to the subsequent probabilistic shaping stage.

3.2. Probabilistic shaping to $\mathcal{N}(0, 1)$. The purpose of probabilistic shaping is not merely to transform discrete symbols into continuous latent values, but to preserve statistical compatibility with the standard Gaussian prior assumed by latent diffusion models. This prior compatibility is important for both stable forward generation and reliable reverse recovery.

We adopt a dithered inverse-CDF shaping strategy to align the symbol sequence with the LDM latent prior. Partition $[0, 1)$ into M equal intervals $I_i = [i/M, (i+1)/M)$. For each $s_n \in \{0, \dots, M-1\}$, draw an i.i.d. dither $u_n \sim \mathcal{U}[0, 1)$ and define

$$\begin{aligned} p_n &:= \frac{s_n + u_n}{M}, \\ \tilde{p}_n &:= \text{clip}(p_n, \varepsilon, 1 - \varepsilon), \\ z_n &:= \Phi^{-1}(\tilde{p}_n), \end{aligned} \quad (13)$$

where Φ and Φ^{-1} denote the CDF and inverse CDF of $\mathcal{N}(0, 1)$, respectively, $\text{clip}(\cdot)$ clamps each value into $[\varepsilon, 1 - \varepsilon]$, and ε is a small constant (default 10^{-7}) used to avoid numerical instability near the CDF tails. Since $p_n \sim \mathcal{U}[0, 1)$, the resulting latent value satisfies $z_n \sim \mathcal{N}(0, 1)$ up to the small bias introduced by clipping.

The shaped sequence $\mathbf{z} = [z_1, \dots, z_N]$ is then written elementwise into the selected latent positions of the LDM tensor in a fixed deterministic order. In this sense, probabilistic shaping provides the transmission-side counterpart of interval decoding, since the same prior-compatible partition of the latent domain is later reused for symbol recovery. The dithers $\{u_n\}$ are ephemeral and need not be stored.

3.3. Stego generation and exact inversion-based latent recovery. On the transmission side, the shaped latent representation is injected into the latent diffusion model and advanced by the forward ODE sampler $\mathcal{A}$, after which the decoder D produces the stego image. On the receiver side, the possibly degraded stego image is first mapped to an initial latent estimate by the encoder E . To reduce the reconstruction error caused by the non-exact encoder-decoder pair in latent diffusion models, this latent estimate is further refined by decoder inversion, which minimizes the reconstruction discrepancy between the received stego image and the image decoded from the optimized latent. The refined latent is then used as the starting point of the diffusion inversion operator $\mathcal{A}^{-1}$, yielding an estimate $\hat{\mathbf{z}}$ of the message-carrying latent representation.

In this work, exact inversion is adopted as the numerical module for latent recovery. Its role is not only to reduce cumulative inversion error along the reverse trajectory, but also to provide a more stable latent estimate for the subsequent interval decoding stage. Following the inversion background introduced in Section 2, the recovery process is implemented by a backward-Euler-based implicit correction with high-order term approximation. For each macro-interval, implicit updates are iteratively performed until convergence, so that the recovered latent remains numerically consistent with the forward sampling process.

Compared with diffusion-based steganographic schemes that mainly use inversion as a recovery tool, the proposed framework explicitly couples inversion fidelity with latent prior compatibility and symbol decision robustness. In other words, the recovered latent is not interpreted in isolation; rather, it is jointly constrained by the prior-compatible shaping rule on the transmission side and the interval decoding rule on the receiver side. Matching the latent representation to the Gaussian prior is expected to reduce statistical mismatch during both generation and recovery.

3.4. Inverse mapping and information recovery. Because the transmission-side shaping maps discrete symbols to prior-compatible latent intervals, message recovery on the receiver side is naturally formulated as an interval-decision problem. This interval decoding mechanism is especially important when $M > 2$, where direct thresholding is no longer sufficient for reliable symbol recovery.

Given each recovered latent element $\hat{z}_n$, compute

$$\begin{aligned} t_n &= \text{clip}(\Phi(\hat{z}_n), \varepsilon, 1 - \varepsilon), \\ \hat{s}_n &= \min\{M - 1, \max\{0, \lfloor M t_n \rfloor\}\}, \end{aligned} \quad (14)$$

where $\lfloor \cdot \rfloor$ denotes the floor operator. Each recovered symbol $\hat{s}_n$ is then converted back to its k -bit binary representation, and the recovered rows are concatenated in order. Finally, the last L_{pad} padding bits are removed to obtain the reconstructed bitstream $\hat{\mathbf{b}}$.

Together with probabilistic shaping, this decoding rule forms a paired mechanism in which the interval structure used for symbol generation is preserved and reused for symbol decision after latent recovery.

3.5. Capacity and complexity. Let $M = 2^k$, and let N denote the number of latent scalars used to carry shaped symbols. Since each symbol represents k bits, the gross payload is

$$B_{\text{gross}} = kN \quad (15)$$

After removing the tail padding of length L_{pad} , the net payload per image is

$$B_{\text{net}} = kN - L_{\text{pad}} \text{ bits} \quad (16)$$

When the entire $C \times H \times W$ latent tensor is fully utilized in an elementwise manner, one has $N = CHW$; more generally, N corresponds to the number of selected latent elements if only a subset is used.

Both shaping and interval-based demapping operate independently on each latent element and therefore have $O(N)$ computational complexity. In practice, however, the dominant computational cost comes from the LDM sampler and, in particular, from the inversion stage rather than the symbol mapping itself. Compared with direct one-pass decoding schemes, the proposed recovery-oriented design introduces additional inversion overhead in exchange for improved latent recovery fidelity and more stable message extraction.

4. Experiments.

4.1. Experimental setup.

4.1.1. *Datasets.* We evaluate the proposed method on two public image-text datasets, namely Flickr8K [17] and the val2017 split of MS-COCO [18]. Flickr8K contains 8,092 images with corresponding captions and is suitable for prompt-conditioned generation with relatively concise semantic descriptions. MS-COCO val2017 contains 5,000 natural images, each associated with five captions, and is adopted as the main benchmark for quantitative evaluation due to its broader visual diversity. Unless otherwise specified, one caption is randomly selected for each image during testing.

4.1.2. *Backbones and compared methods.* To examine whether the proposed transmission-recovery framework remains effective across different Stable Diffusion backbones, we evaluate our method on Stable Diffusion v1.5 (SD v1.5) [19] and Stable Diffusion v2.1 (SD v2.1) [20]. For both backbones, the image resolution is fixed at 512×512 , and the latent size is $4 \times 64 \times 64$.

We retain three representative diffusion-based generative steganography baselines, namely GRDH [14], StegaMGM [15], and StegaDDPM [13]. GRDH is adopted as the most directly related robust mapping-based latent-diffusion baseline. StegaMGM is included as a recent SD v2.1-based generative steganography method with inversion-based recovery. StegaDDPM is retained as an earlier DDPM-based generative steganography baseline. In our comparison, StegaDDPM is referenced at the commonly used operating point of 256×256 with a payload of 4,096 bits, as adopted in subsequent comparative evaluations. StegaMGM reports three payload settings on 512×512 images, namely 16,384, 32,768, and 49,152 bits, corresponding to $p = 1, 2, 3$. GRDH uses a practical payload setting of 16,384 bits on 512×512 images.

For GRDH, we directly use the results reported in the original paper and treat it as the main matched reference baseline. For StegaMGM and StegaDDPM, the entries explicitly reported in the original papers are directly incorporated when the corresponding settings are matched, while the distortion settings not reported in the original papers are supplemented by our own reproduction under their original backbone-related configurations. In particular, the StegaMGM reproduction follows its original SD v2.1-based setting as closely as possible, whereas the StegaDDPM reproduction is used only for the entries unavailable from its original report.

4.1.3. *Implementation details.* Unless otherwise specified, we follow the backbone-related settings of GRDH [14], namely $N = 64$, $C_f = C_r = 4$, $H = W = 512$, $f = 8$, and guidance scale $w = 5.0$, so that the comparison is not affected by changes in the underlying generation configuration. For our method, the generation and inversion processes are both implemented with DPM-Solver++(2M), and the default numbers of sampling and inversion steps are both set to 20. In the security experiment, the sampling step is increased to 50 to generate more stable stego images for steganalysis evaluation.

For message embedding, we consider two symbol cardinalities, i.e., $M = 2$ and $M = 4$, in order to explicitly study the trade-off between embedding capacity and recovery robustness. Here, $M = 2$ corresponds to binary symbol mapping with $k = 1$, while $M = 4$ corresponds to 4-ary symbol mapping with $k = 2$. In probabilistic shaping, the clipping constant is fixed to $\varepsilon = 10^{-7}$ to avoid numerical instability near the tails of the inverse CDF.

For exact inversion, we follow the latent-diffusion setting reported in [21] as closely as possible. In particular, the inversion is conducted in float32, and the micro-step partition number is set to $J = 10$. For decoder inversion, we use Adam with learning rate

0.1 together with a cosine learning rate scheduler, and perform 100 iterations. For the StegaMGM reproduction, we additionally follow its original SD v2.1 setting with guidance scale 3.0 [15]. All experiments are implemented in PyTorch and conducted on four NVIDIA RTX 4090 GPUs.

4.1.4. *Evaluation metrics.* We evaluate the compared methods from five aspects, namely hiding capacity, extraction accuracy, robustness, security, and image visual quality.

1) Hiding Capacity. We measure hiding capacity in bits-per-pixel (bpp), i.e.,

$$\text{bpp} = \frac{B_{\text{net}}}{H \times W}, \quad (17)$$

where B_{net} denotes the net payload per image, and H and W are the image height and width, respectively.

2) Extraction Accuracy. Extraction accuracy is measured by the normalized Hamming distance between the transmitted bitstream $\mathbf{b}$ and the recovered bitstream $\hat{\mathbf{b}}$:

$$\text{Acc} = 1 - \frac{\text{dist}(\mathbf{b}, \hat{\mathbf{b}})}{B}, \quad (18)$$

where $\text{dist}(\cdot, \cdot)$ denotes the Hamming distance and B is the bit length of $\mathbf{b}$.

3) Robustness. To evaluate robustness under lossy transmission, we report extraction accuracy after a set of common post-processing operations. Specifically, we consider a) Resize with scaling factors $\{0.5, 0.75, 1.25, 1.5\}$; b) JPEG compression with quality factors $\{90, 70, 50\}$; c) Median blur with kernel sizes $\{3 \times 3, 5 \times 5, 7 \times 7\}$; d) Gaussian blur with kernel sizes $\{3 \times 3, 5 \times 5, 7 \times 7\}$; and e) additive Gaussian noise with standard deviations $\{0.01, 0.05, 0.10\}$. For comparison, we also report two non-attack cases, namely lossless storage and PNG recompression.

4) Security. In the context of generative steganography, we define cover images as images generated from randomly sampled Gaussian latents, and stego images as images generated from message-conditioned latents. Security is evaluated by the detection error rate

$$P_E = \frac{P_{FA} + P_{MD}}{2}, \quad (19)$$

where P_{FA} and P_{MD} denote the false alarm rate and missed detection rate, respectively. A P_E value close to 0.5 indicates that the detector has limited ability to distinguish stego images from normally generated cover images.

5) Image Visual Quality. We use FID_{real} and BRISQUE to evaluate the visual quality of generated images. FID_{real} measures the distributional discrepancy between generated images and real images from the corresponding dataset. BRISQUE is adopted as a no-reference perceptual image-quality metric. Lower values of both metrics indicate better visual quality.

4.2. Extraction accuracy and robustness. To provide a more complete robustness comparison, we evaluate the retained baseline methods and the proposed method under the same distortion categories considered in this work, including lossless storage, PNG recompression, image resizing, JPEG compression, median filtering, Gaussian filtering, and additive Gaussian noise. For GRDH, the reported numbers are directly taken from the original paper. For StegaMGM and StegaDDPM, the entries explicitly reported in the original papers are directly included when the corresponding settings are matched, while the missing distortion settings are supplemented by our own reproduction under their original backbone-related configurations.

Table 1 summarizes the extraction accuracy under different distortions. Unless otherwise specified, the most directly matched cross-method comparison is the practical payload

TABLE 1. Extraction accuracy (%) under different distortions

Operation	Setting	StegaDDPM	GRDH	StegaMGM ($p = 1$)	Ours (SD v1.5, $M = 2$)	Ours (SD v2.1, $M = 2$)	Ours (SD v1.5, $M = 4$)	Ours (SD v2.1, $M = 4$)
Lossless	—	99.00	98.12	99.63	99.07	99.19	98.33	98.50
PNG	—	50.00	98.05	99.05	98.82	98.89	98.20	98.35
Resize	0.50	51.62	94.86	83.94	96.12	96.37	95.12	95.19
	0.75	60.15	97.03	91.69	97.86	98.04	96.85	97.02
	1.25	72.09	97.74	94.00	98.34	98.49	96.23	96.79
	1.50	75.36	97.89	96.45	98.28	98.58	96.05	96.85
JPEG quality	90	50.00	96.34	99.41	97.88	98.89	95.21	95.93
	70	50.00	92.35	98.38	94.54	96.31	90.82	91.73
	50	50.00	88.87	96.65	92.12	94.23	87.69	89.01
Median blur	3×3	56.00	95.84	88.95	97.64	97.89	94.65	95.10
	5×5	52.00	90.10	84.02	92.22	93.35	88.43	89.23
	7×7	51.00	82.84	82.46	87.59	90.34	80.78	81.56
Gaussian blur	3×3	50.00	97.12	89.07	98.24	98.87	96.31	96.85
	5×5	50.00	95.79	83.23	96.87	97.29	93.02	93.76
	7×7	50.00	93.12	81.86	95.56	96.23	90.67	91.45
Gaussian noise	0.01	62.00	97.38	99.36	98.37	98.62	96.57	97.12
	0.05	52.00	90.94	97.07	93.23	93.84	90.02	90.64
	0.10	50.00	84.73	94.26	87.46	87.79	80.39	81.14

GRDH results are directly reported in the original paper. For StegaMGM and StegaDDPM, the entries explicitly available in the original papers are directly included when matched, while the distortion settings not reported in the original papers are supplemented by our own reproduction under their original backbone-related configurations.

level of 0.0625 bpp, i.e., GRDH, StegaMGM ($p = 1$), and our $M = 2$ setting. The $M = 4$ results are additionally reported to illustrate how robustness changes under a higher payload level of 0.1250 bpp. This design allows us to jointly examine robustness differences caused by both the generative backbone and the payload level.

As shown in Table 1, the proposed method consistently outperforms GRDH under the tested distortions at the matched practical payload level of 0.0625 bpp, while showing competitive performance with StegaMGM under several common storage and perturbation conditions. As expected, extraction accuracy gradually decreases as the distortion strength increases. Nevertheless, the proposed framework remains robust over a broad range of common perturbations. In addition, the comparison between SD v1.5 and SD v2.1, together with the results of $M = 2$ and $M = 4$, allows us to jointly examine the influence of the generative backbone and the payload level on message recovery.

4.3. Hiding capacity. We next compare the payload scale of the retained baselines and the proposed method. In the proposed method, the latent tensor contains $4 \times 64 \times 64 = 16,384$ elements. Therefore, the net payloads are

$$B_{\text{net}}^{(M=2)} = 16,384 \text{ bits}, \quad B_{\text{net}}^{(M=4)} = 32,768 \text{ bits}, \quad (20)$$

which correspond to

$$\text{bpp}_{M=2} = \frac{16,384}{512 \times 512} = 0.0625, \quad (21)$$

and

$$\text{bpp}_{M=4} = \frac{32,768}{512 \times 512} = 0.1250. \quad (22)$$

Table 2 shows that our $M = 2$ and $M = 4$ settings align exactly with the first two payload levels of StegaMGM in terms of bpp, while substantially exceeding the commonly used payload of StegaDDPM in absolute size. GRDH uses a single practical payload setting of 0.0625 bpp, which corresponds to 16,384 bits on 512×512 images [14]. Taken

TABLE 2. Payload comparison of the retained baselines and the proposed method

Method	Image size	Payload	bpp
StegaDDPM [13]	256×256	4,096	0.0625
GRDH [14]	512×512	16,384	0.0625
StegaMGM ($p = 1$) [15]	512×512	16,384	0.0625
StegaMGM ($p = 2$) [15]	512×512	32,768	0.1250
StegaMGM ($p = 3$) [15]	512×512	49,152	0.1875
Ours ($M = 2$)	512×512	16,384	0.0625
Ours ($M = 4$)	512×512	32,768	0.1250

together, these results show that the proposed framework supports an explicit capacity extension from 0.0625 bpp to 0.1250 bpp within a unified embedding and recovery framework.

4.4. Image visual quality. Figures 2 and 3 show representative stego images generated by the proposed framework under different prompts using Stable Diffusion v1.5 and v2.1,



FIGURE 2. Representative stego images generated under different prompts using Stable Diffusion v1.5



FIGURE 3. Representative stego images generated under different prompts using Stable Diffusion v2.1

respectively. Overall, the generated images remain visually natural and semantically consistent with the corresponding prompts for both backbones, indicating that latent-domain message transmission does not noticeably undermine controllable text-to-image synthesis. The visual examples also suggest that the proposed framework can be effectively applied to different latent-diffusion backbones while preserving favorable perceptual quality.

To quantitatively assess visual quality, we report FID_{real} and BRISQUE on two datasets, namely MS-COCO and Flickr8K. For each dataset, FID_{real} measures the distributional discrepancy between the generated images and the corresponding real-image set, while BRISQUE is adopted as a no-reference perceptual quality metric to evaluate the visual quality of the resulting images.

For the SD v1.5 setting, the values of FID_{real} and BRISQUE for GRDH and the proposed method follow the original visual-quality evaluation protocol on MS-COCO and Flickr8K, where natural images from the two datasets are used as references. The StegaMGM entries and the SD v2.1 results are evaluated under the same dataset-level protocol in our implementation and are also reported in Table 3.

TABLE 3. Visual-quality comparison on MS-COCO and Flickr8K. Lower values indicate better quality.

Method	MS-COCO		Flickr8K	
	$FID_{\text{real}} \downarrow$	BRISQUE $\downarrow$	$FID_{\text{real}} \downarrow$	BRISQUE $\downarrow$
GRDH [14]	20.5693	19.4550	16.6004	21.3189
StegaMGM ($p = 1$) [15]	20.5352	20.2502	16.7153	21.8236
Ours (SD v1.5, $M = 2$)	21.2678	20.2672	16.9826	22.2924
Ours (SD v2.1, $M = 2$)	20.4717	19.7220	16.6211	21.6804
Ours (SD v1.5, $M = 4$)	20.5421	19.5845	16.4710	21.5603
Ours (SD v2.1, $M = 4$)	20.3035	19.4254	16.3104	21.3243

The results on MS-COCO and Flickr8K jointly reflect the visual fidelity of the generated stego images from both distributional and perceptual perspectives. In particular, FID_{real} evaluates how close the generated images are to the corresponding real-image distribution, whereas BRISQUE complements this analysis by assessing perceptual quality without requiring reference images. The results indicate that the proposed method preserves favorable visual quality on both datasets. In addition, within the proposed framework, the 4-ary setting ($M = 4$) consistently yields better visual quality than the binary setting ($M = 2$), which reflects the trade-off between visual quality and the other performance metrics such as robustness and payload.

4.5. Security. To evaluate resistance against learned steganalysis, we use four representative steganalysis networks, namely SRNet [9], Ye-Net [22], XuNet [23], and SiaStegNet [24]. Following our original security-evaluation protocol, we conduct the steganalysis experiment on MS-COCO. Specifically, 4,000 cover images and 4,000 stego images are

TABLE 4. Steganalysis resistance of the proposed method

Method	SRNet	Ye-Net	XuNet	SiaStegNet
Ours (SD v1.5, $M = 2$)	0.5035	0.5233	0.5141	0.4867
Ours (SD v1.5, $M = 4$)	0.4940	0.4910	0.5350	0.5133
Ours (SD v2.1, $M = 2$)	0.5070	0.4955	0.4915	0.5108
Ours (SD v2.1, $M = 4$)	0.4994	0.5038	0.5012	0.4975

used for training, while 500 cover images and 500 stego images are used for testing. To obtain stable stego images for steganalysis evaluation, the sampling step is set to $S = 50$ in this subsection. The corresponding results are summarized by the detection error rate P_E .

A P_E value close to 0.5 indicates that the detector has limited ability to distinguish stego images from normally generated cover images. These results suggest that the proposed framework maintains low steganographic detectability under the evaluated settings.

5. Conclusion. In this paper, we investigated reliable message recovery in latent-diffusion-based generative image steganography. To address the coupled challenges of latent prior mismatch and cumulative inversion error, we developed a recovery-oriented framework that integrates probabilistic-shaping-guided bit-to-latent mapping, exact inversion for DPM-Solver++(2M), and interval decoding for robust symbol recovery. Under this framework, secret bits are embedded into the latent generation process while remaining compatible with the Gaussian prior of latent diffusion models. Experimental results demonstrate that the proposed method achieves more stable message extraction under common lossy perturbations while maintaining competitive image quality and low steganographic detectability. The comparison between two symbol cardinalities, i.e., $M = 2$ and $M = 4$, further highlights the intrinsic trade-off between embedding capacity and recovery robustness: higher-capacity embedding can be achieved, but at the cost of increased sensitivity to perturbations.

Several limitations remain. The current study is mainly conducted on Stable Diffusion with DPM-Solver++(2M), and its generalization to other latent-diffusion or flow-based generative models requires further investigation. In addition, although the proposed method is robust to common lossy distortions, its behavior under stronger semantic editing, regeneration, and cross-model inversion scenarios has not yet been systematically examined.

Future work will focus on extending the proposed framework to more advanced generative backbones and inversion schemes, improving high-payload recovery through more powerful coding and decoding strategies such as soft-decision and error-correcting mechanisms, and evaluating robustness under more realistic image editing and dissemination conditions. More broadly, the recovery-oriented perspective developed in this work may also provide a useful basis for studying reliable information recovery in other generative information hiding settings.

REFERENCES

- [1] B. Song, P. Wei, S. Wu, Y. Lin and W. Zhou, A survey on deep-learning-based image steganography, *Expert Systems with Applications*, vol.254, Article no.124390, 2024.
- [2] N. J. De La Croix, T. Ahmad and F. Han, Comprehensive survey on image steganalysis using deep learning, *Array*, vol.22, Article no.100353, 2024.
- [3] K. Hu, M. Wang, X. Ma, J. Chen, X. Wang and X. Wang, Learning-based image steganography and watermarking: A survey, *Expert Systems with Applications*, vol.249, Article no.123715, 2024.
- [4] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, Generative adversarial networks: An overview, *IEEE Signal Processing Magazine*, vol.35, no.1, pp.53-65, 2018.
- [5] Y. Cao, Z. Zhou, Q. Wu, C. Yuan and X. Sun, Coverless information hiding based on the generation of anime characters, *EURASIP Journal on Image and Video Processing*, vol.2020, no.1, pp.1-15, 2020.
- [6] Z. Zhang, G. Fu, R. Ni, J. Liu and X. Yang, A generative method for steganography by cover synthesis with auxiliary semantics, *Tsinghua Science and Technology*, vol.25, no.4, pp.516-527, 2020.
- [7] D. Hu, L. Wang, W. Jiang, S. Zheng and B. Li, A novel image steganography method via deep convolutional generative adversarial networks, *IEEE Access*, vol.6, pp.38303-38314, 2018.

- [8] J. Li, K. Niu, L. Liao, L. Wang, J. Liu, Y. Lei and M. Zhang, A generative steganography method based on WGAN-GP, *Proc. of International Conference on Artificial Intelligence and Security*, pp.386-397, 2020.
- [9] M. Boroumand, M. Chen and J. Fridrich, Deep residual network for steganalysis of digital images, *IEEE Transactions on Information Forensics and Security*, vol.14, no.5, pp.1181-1193, 2018.
- [10] Z. Zhou, Y. Su, J. Li, K. Yu, Q. M. J. Wu, Z. Fu and Y. Shi, Secret-to-image reversible transformation for generative steganography, *IEEE Transactions on Dependable and Secure Computing*, vol.20, no.5, pp.4118-4134, 2023.
- [11] D. P. Kingma and P. Dhariwal, GLOW: Generative flow with invertible 1×1 convolutions, *Advances in Neural Information Processing Systems*, vol.31, 2018.
- [12] J. Ho, A. Jain and P. Abbeel, Denoising diffusion probabilistic models, *Advances in Neural Information Processing Systems*, vol.33, pp.6840-6851, 2020.
- [13] Y. Peng, D. Hu, Y. Wang, K. Chen, G. Pei and W. Zhang, StegaDDPM: Generative image steganography based on denoising diffusion probabilistic model, *Proc. of the 31st ACM International Conference on Multimedia*, Ottawa, Canada, pp.7143-7151, 2023.
- [14] X. Hu, S. Li, Q. Ying, W. Peng, X. Zhang and Z. Qian, Establishing robust generative image steganography via popular stable diffusion, *IEEE Transactions on Information Forensics and Security*, vol.19, pp.8094-8108, 2024.
- [15] J. Jiang, Z. Wang, Z. Yuan and X. Zhang, Generative image steganography based on text-to-image multimodal generative model, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.35, no.9, pp.8907-8916, 2025.
- [16] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li and J. Zhu, DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models, *Machine Intelligence Research*, vol.22, no.4, pp.730-751, 2025.
- [17] M. Hodosh, P. Young and J. Hockenmaier, Framing image description as a ranking task: Data, models and evaluation metrics, *Journal of Artificial Intelligence Research*, vol.47, pp.853-899, 2013.
- [18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, Microsoft COCO: Common objects in context, *Proc. of European Conference on Computer Vision*, pp.740-755, 2014.
- [19] R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, High-resolution image synthesis with latent diffusion models, *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.10684-10695, 2022.
- [20] Stability AI, *Stable Diffusion 2.1*, <https://huggingface.co/stabilityai/stable-diffusion-2-1>, 2022.
- [21] S. Hong, K. Lee, S. Y. Jeon, H. Bae and S. Y. Chun, On exact inversion of DPM-solvers, *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.7069-7078, 2024.
- [22] J. Ye, J. Ni and Y. Yi, Deep learning hierarchical representations for image steganalysis, *IEEE Transactions on Information Forensics and Security*, vol.12, no.11, pp.2545-2557, 2017.
- [23] G. Xu, H.-Z. Wu and Y.-Q. Shi, Structural design of convolutional neural networks for steganalysis, *IEEE Signal Processing Letters*, vol.23, no.5, pp.708-712, 2016.
- [24] W. You, H. Zhang and X. Zhao, A siamese CNN for image steganalysis, *IEEE Transactions on Information Forensics and Security*, vol.16, pp.291-306, 2021.

Author Biography



Wanting Sun received the B.Eng. degree in Electronic Information Engineering from the School of Electronic and Information Engineering, Tiangong University, China, in 2019. She is currently pursuing the Ph.D. degree in Aeronautical and Astronautical Science and Technology at the School of Aeronautics and Astronautics, Zhejiang University, China. Her research interests include image information hiding and multimedia information security.



Zhe-Ming Lu received the B.S. and M.S. degrees in Electrical Engineering and the Ph.D. degree in Measurement Technology and Instrumentation from the Harbin Institute of Technology (HIT), China, in 1995, 1997, and 2001, respectively. He became a Lecturer with HIT in 1999. Since 2003, he has been a Professor with the Department of Automatic Test and Control, HIT. He is currently a Full Professor with the School of Aeronautics and Astronautics, Zhejiang University, China. In the areas of multimedia signal processing and information hiding, he has published more than 350 papers, seven monographs in Chinese, two monographs in English, and three book chapters in English. His current research interests include multimedia signal processing, information security, and complex networks.