PSyDUCK: Hiding Information in the Denoising Process of Latent Diffusion Models

1st Aqib Mahfuz*
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
aqib.mahfuz@gmail.com

2nd Georgia Channing*
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
cgeorgia@robots.ox.ac.uk

3rd Mark van der Wilk
Department of Computer Science
University of Oxford
Oxford, United Kingdom
mark.vdwilk@cs.ox.ac.uk

4th Philip H.S. Torr
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
philip.torr@eng.ox.ac.uk

5th Fabio Pizzati
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
fabio.pizzati@eng.ox.ac.uk

6th Christian Schroeder de Witt
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
cs@robots.ox.ac.uk

***Abstract*—Recent advances demonstrate that information can be covertly embedded in the outputs of stochastic generative AI models, raising both opportunities for secure communication and risks of misuse. Existing latent diffusion steganography methods typically hide data in the entropy of the initial latent state, inherently limiting embedding capacity. In this work, we instead investigate information hiding within the entropy of the diffusion denoising process itself. We introduce PSyDUCK, a simple but efficient framework that leverages controlled divergence and local mixing during denoising to enable high-capacity message embedding while preserving visual fidelity. Our empirical evaluation shows PSyDUCK can hide substantial information in both image and video diffusion models. While our formal analysis indicates that the security guarantees of denoising-based embedding are limited, the existence of this channel nonetheless requires that steganalysis methods account for entropy throughout the entire denoising process - not just in the initial latent state.**

***Index Terms*—steganography, information hiding, latent diffusion models, privacy**

I. Introduction

Steganography, the art of hiding information within innocuous media, has long enabled covert communication in under surveillance for over a thousand year. As AI-generated content proliferates the internet, methods have been developed that allow for new and scalable opportunities for steganographic embedding in generative diffusion model [1] outputs. These approaches to diffusion model steganography have focused on modifying pixel spaces [2] or initial latent state entropy [3], [4]. However, these methods face capacity limitations and often struggle to adapt to modern latent diffusion architectures.

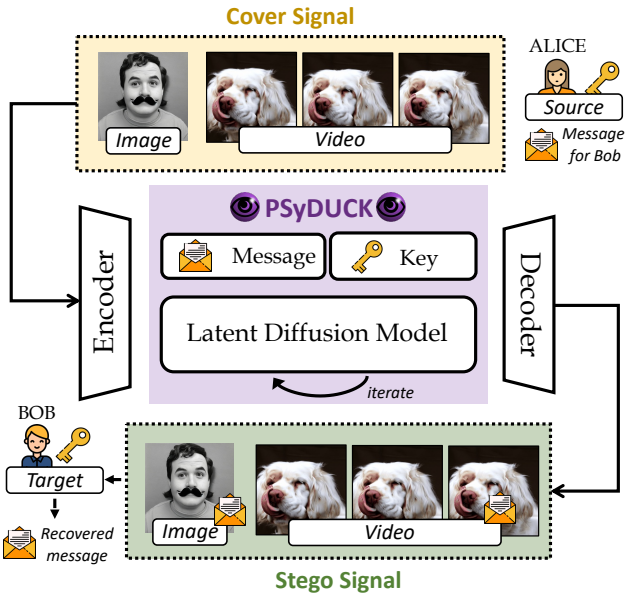


Fig. 1: **General PSyDUCK scheme.** Alice wants to send a secret message to Bob while preventing interception by malicious actors. To achieve this, PSyDUCK embeds a steganographic message of arbitrary length into a *cover signal*–an image or video–using shared keys and pre-trained latent diffusion models. The resulting *stego-signal* can be freely shared on the open web, allowing Bob, who possesses the correct key, to decode and retrieve the original message.

In this work, we introduce *Practical Steganography via Diffusion Using Construction Keys* (PSyDUCK), a model-agnostic steganographic framework that, for the first time, leverages the entropy of a multi-step latent denoising process itself for controlled, robust message embedding and recovery. Just as [2], [4], PSyDUCK assumes sender and receiver share access to a public prompt-conditioned latent diffusion

model and secret keys. PSyDUCK harnesses the entropy of the denoising process across multiple steps by controlling divergence and local mixing. In doing so, PSyDUCK overcomes previous capacity bottlenecks, enabling high-fidelity, high-capacity steganography for both images and videos without the need to retrain or or fine-tune the underlying diffusion model.

Our empirical evaluations demonstrate that PSyDUCK can efficiently hide substantial information in both image and video diffusion models. This significantly increases the available entropy for information hiding, in particular as PSyDUCK could be employed on top of existing schemes [3], [4]. Despite our formal analysis reveals that PSyDUCK's security is bounded by properties of its learnt diffusion function, we empirically find its outputs to maintain visual quality. This both opens up a new avenue for research on information hiding in latent diffusion models, as well as underscores the need for steganography detectors to not only focus on the entropy in the initial latent, but also the wider diffusion process.

A. Summary of Contributions

- We present the first general, training-free method for steganography in latent diffusion models that exploits the denoising process, supporting both image and video modalities.
- We analyze PSyDUCK's error and detection properties, highlighting both its practical capacity and imperceptibility, and its limitations in theoretical secrecy.
- We benchmark PSyDUCK against existing methods, demonstrating superior scalability and adaptability.

Overall, PSyDUCK expands the threat model for AI-enabled steganography and provides crucial guidance for the development of future steganalysis techniques.

II. Related Work

Traditional image steganography focuses on embedding visually imperceptible perturbations into a chosen cover image, for example using least significant bit encoding [5] or deep neural network encoders [6], are generally susceptible to statistical detection [7].

Generative steganography instead embeds hidden information in the sampling process of a generative AI model. To remain undetected, steganography both needs to maintain innocuousness of the used communications channel (*covertext*) relative to the sea of other communication channels, as well as keep the *stegotext* distribution, i.e. the distribution of encoded channel content, indistinguishable from the chosen channel's covertext distribution. As generic pre-trained foundation models are producing increasingly realistic content, and these models are widely used for producing AI-generated content, training-free steganography methods that can be deployed on top of pre-trained foundation models and aim to maintain their output distribution, are of particular interest. While perfect secrecy [8] has only been shown to exist for autoregressive generative models in the training-free setting, provably secure steganography has recently been established in pixel diffusion [2] and latent diffusion models [4]. All currently known steganographic methods for diffusion models that do not require sender and receiver to share model prompts are deemed insecure against polynomially-bounded attackers [3]. As an example of a non-training free method, [9] proposes retraining a diffusion model for the express purpose of steganography.

III. Preliminaries

Generative diffusion models, inspired by non-equilibrium thermodynamics, have emerged as a powerful framework for generating complex data distributions from simple noise [10]. These models operate through a probabilistic process consisting of a forward diffusion and a reverse denoising process.

The forward diffusion process incrementally adds Gaussian noise to an initial data point $\mathbf{x}_0 \in \mathbb{R}^l$ over T steps, producing a sequence $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. Each step is modeled as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where β_t is the variance schedule that controls the noise level at step t. By the final step, $\mathbf{x}_T$ approximates a standard Gaussian distribution [10].

The reverse denoising process learns to transform noisy data back into the original data distribution by iteratively removing the added noise. A neural network is trained to predict the noise component at each step, enabling the recovery of $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$. The reverse process at each step can be modeled as:

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_\theta^2(t)\mathbf{I}), \tag{2}$$

where $\mu_\theta(\mathbf{x}_t, t)$ represents the predicted mean and $\sigma_\theta^2(t)$ is the predicted variance, both parameterized by the neural network. During this process, the noise ϵ_t is sampled from the Gaussian distribution defined by $\mu_\theta(\mathbf{x}_t, t)$ and $\sigma_\theta^2(t)$ as:

$$\epsilon_t \sim \mathcal{N}(0, \sigma_\theta^2(t)). \tag{3}$$

The sampled noise is then used to adjust the latent variable in the denoising step.

Conditional diffusion models extend the generative framework by incorporating additional information (e.g., class labels or textual descriptions) to guide the generation process [11].

IV. The PSyDUCK Framework

Let T be the total number of denoising steps of a latent diffusion model. We assume that Alice (the sender) and Bob (the receiver) share **(1)** a *synchronisation key* k_s, **(2)** a small set of *reference keys* $\{k_i\}_{i=0}^{r-1}$ with $r \geq 2$, and **(3)** the public parameters of the diffusion model (noise schedule $\{\beta_t\}_{t=1}^{T}$, network weights, encoder/decoder, *etc.*). Each key is realised as a conditioning vector or prompt embedding that guides the reverse process. In our experiments we simply use different random seeds, but any prompt-conditioning mechanism compatible with the diffusion network can serve as a key. The message to be transmitted is a bit-string $\mathbf{b} \in \{0,1\}^\ell$ whose length ℓ depends on the number of selectable units (pixels or latent patches) available during the last d denoising steps. A general architecture diagram is presented in Figure 2.

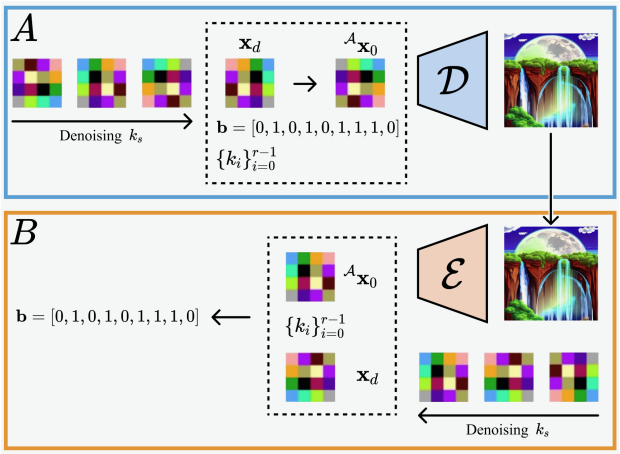
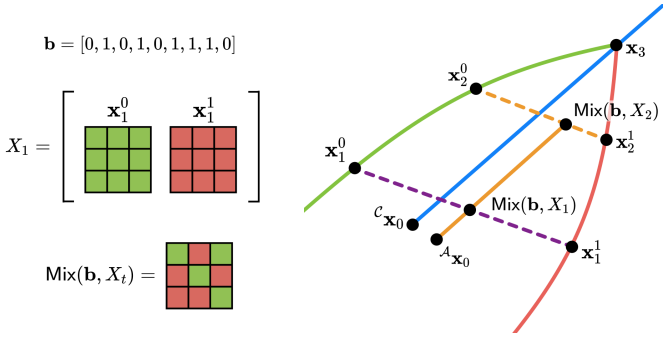


Fig. 2: **Encoding and Decoding.** An illustration of the PSyDUCK encoding and decoding processes on latent model architectures. Dashed boxes denote custom PSyDUCK operations. To encode secret bitstring $\mathbf{b}$, Alice first denoises in the latent space until timestep d with synchronization key k_s. Then, she diverges for d steps using reference keys $\{k_i\}_{i=0}^{r-1}$ and subsequently mixes the diverged samples using $\mathbf{b}$. Finally, she puts her sample through the decoder $\mathcal{D}$ to transmit her final output, ${}^{\mathcal{A}}\mathbf{x}_0$. To extract $\mathbf{b}$, Bob first projects Alice's transmission back into the latent space with encoder $\mathcal{E}$. He then repeats her denoising trajectory until timestep d with synchronization key k_s. Then, like Alice, he diverges for d steps using the reference keys $\{k_i\}_{i=0}^{r-1}$. Bob finally decodes Alice's message $\mathbf{b}$ by comparing his reference samples to Alice's transmission.



(a) The Mix operation. (b) Diverging and mixing.

Fig. 3: **Diverging and Mixing.** a An example of mixing the red and green trajectories based on $\mathbf{b}$ to form a mixed sample when $t = 1$ and $r = 2$. b Here, the blue path represents the original trajectory of the diffusion model, resulting in the cover-image ${}^{\mathcal{C}}\mathbf{x}_0$. The green path represents the divergent trajectory when conditioned with k_0, while the red path presents the divergent trajectory when conditioned with k_1. Orange: Alice's trajectory upon mixing samples from the two divergent paths.

a) Two-Phase Denoising Process: Our steganographic protocol, dubbed PSyDUCK, splits the reverse diffusion into two phases: synchronized denoising and branched denoising.

In the synchronized denoising phase, from $t = T$ down to $t = d+1$, both parties run the usual reverse process conditioned on the shared key k_s. Because the same key and network weights are used, the latent trajectory is identical for Alice and Bob up to numerical precision.

At $t = d$ the process forks and the branched denoising begins: Alice (resp. Bob) instantiates r copies of the current latent, denoises each copy for another step with a different reference key k_i, and collects the resulting *reference samples* $\{\mathbf{x}_t^i\}_{i=0}^{r-1}$. During the remaining d steps ($t = d,\ d-1, \ldots, 1$), each reference sample evolves independently under its own key, thus spanning r well-separated but deterministic trajectories in latent space.

b) Encoding: After denoising to $t = 1$, Alice combines the set of r reference latents $X_1 = [\mathbf{x}_1^0, \ldots, \mathbf{x}_1^{r-1}]$ into a single latent $\mathbf{x}_1$ using a deterministic mixing function:

$$ {}^{\mathcal{A}}\mathbf{x}_1 = \mathsf{Mix}\big(\mathbf{b}, [\mathbf{x}_1^0, \ldots, \mathbf{x}_1^{r-1}]\big) \tag{4}$$

For $r = 2$, illustrated in Figure 3a, the operator reduces to a bit-wise switch $\mathsf{Mix}(\mathbf{b}, [\mathbf{x}_1^0, \mathbf{x}_1^1])[j] = \big(\mathbf{x}_1^{\mathbf{b}[j]}\big)[j]$ where the index j runs over pixels (pixel-space model) or spatial tokens (latent model). A graphical illustration of the branched denoising and mixing processes where $r = 2$ is shown in Figure 3b. If $r > 2$, we can generalize by mapping each bit-chuck $\mathbf{b}[j]$ to an integer in $\{0, \ldots, r-1\}$ and mixing with the corresponding reference sample. Finally, Alice takes a final denoising step, from $\mathbf{x_1}$ to $\mathbf{x_0}$, with the original key k_s used in the synchronized denoising phase, decodes the mixed latent into the pixel-space, and yields the stego-image to Bob.

c) Decoding: Receiving the stego-image, Bob repeats the synchronized denoising phase exactly, reproducing the shared latent $\mathbf{x}_d$. From $t = d$ downward he again spawns r reference trajectories using the keys $\{k_i\}_{i=0}^{r}$. At $t = 0$ he holds the r candidate reconstructions $\{\mathbf{x}_0^i\}_{i=0}^{r-1}$ as well as Alice's transmitted latent ${}^{\mathcal{A}}\mathbf{x}_0$. For every selectable unit j he computes the absolute distance $\mathbf{d}_i[j] = \big|\mathbf{x}_0^i[j] - {}^{\mathcal{A}}\mathbf{x}_0[j]\big| \quad (i = 0, \ldots, r-1)$ and assigns the bit $\hat{\mathbf{b}}[j] = \arg\min_i \mathbf{d}_i[j]$.

V. Theoretical Analysis

We now analyse the security of our framework. Specifically, we are considering whether the *stegosystem* induced by PSyDUCK is *steganographically secret against chosen hiddentext attacks* [12]. This can be viewed as analogous to IND$-CPA in cryptography but with respect to the covertext distribution $\mathcal{C}$, which might not be random uniform.

We make the following assumptions about message preparation. To prevent forgery attacks [13], we assume that the sender applies a private signature to each message that can be authenticated its messages using a public verification key as in [4]. The signed message is then embedded into a random uniform distribution. This can be achieved through XOR-ing the message with a fresh *one-time pad (OTP)* shared between sender and receiver [8], or - up to IND$-CPA security - using a suitable public-key encryption scheme with a fresh random seed [14]. This uniform embedding prevents a conditioning of the stegotext on the (assumed unknowable) message distribution and prevents stegotext distortion under message repetition [3].

Proposition 1. *PSyDUCK is not provably secure under arbitrary diffusion models and denoising schedules for $d \geq 2$.*

Abridged. Let $r = d = l = 2$ with independently chosen secret keys k_1, k_2, and a binary message $m \in \{00, 01, 10, 11\}$ chosen uniform at random. Then, pre-encoding, the latents $j = 1, 2$ in the output layer (d=0) and assuming identical starting latents $\mathbf{x}_2^1 = \mathbf{x}_2^2$ are given by

$$\mathbf{x}_j^0 = \mu_{\theta,1}\left(\mu_{\theta,2}(\mathbf{x}_2^j, 2) + \epsilon_2^j\right) + \epsilon_1^j \tag{5}$$

with $\epsilon_{t-1}^j \overset{k_j}{\sim} \mathcal{N}(0I, \sigma(t)I)$ where noise terms are assumed uncorrelated by construction, i.e. $\mathbb{E}[\epsilon_t^j \epsilon_{t'}^k] = 0 \quad \forall j, k, t, t' \in [0, 1]$ (see Section IV). Embedding m with Mix then results in the stegotext $\tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}}[i] = \mathbf{x}_{m[i]}^0[i]$.

We now show that $\exists \mu_\theta(\cdot, 1)$ such that the average total variation distance $\mathbb{E}_m\left[\text{TVD}\left(\mathbb{P}(\tilde{\mathbf{x}}), \mathbb{P}(\mathbf{x}_0^1)\right)\right] > 0$. Specifically, assume $\mu_\theta(\cdot, 1) = \langle 1, 0\rangle$ if $\text{sgn}(\mathbf{x}_0^1)[0] = \text{sgn}(\mathbf{x}_0^1)[1]$, and $\langle 0, 1\rangle$ otherwise. Assuming that $\sigma(0) \to 0$ and that $\mu_\theta(\cdot, \cdot)$ and $\mathbf{x}_2^1$ result in $\mathbf{x}_1^1$ that is distributed such that all possible outcomes of $\mu_\theta(\cdot, 1)$ have non-zero support. Then clearly, whenever $m \in \{01, 10\}$, $\text{TVD}\left(\mathbb{P}(\tilde{\mathbf{x}}), \mathbb{P}(\mathbf{x}_0^1)\right) > 0$, and, in the worst case, $\mathbb{E}_m\left[\text{TVD}\left(\mathbb{P}(\tilde{\mathbf{x}}), \mathbb{P}(\mathbf{x}_0^1)\right)\right] = \frac{1}{4}$.

We have thus already shown that perfect secrecy cannot be guaranteed under arbitrary μ_θ. Furthermore, by Hoeffding's inequality, $\mathbb{E}_m\left[\text{TVD}\left(\mathbb{P}(\tilde{\mathbf{x}}), \mathbb{P}(\mathbf{x}_0^1)\right)\right]$ can be empirically estimated with an exponentially-decaying error term in the number of samples, therefore rendering the scheme insecure even under polynomially-bounded attackers. This conclusion generalises by induction to $r, d, l > 2$ and longer messages. □

Despite the lack of statistical security guarantees, however, PSyDUCK does produce outputs with high visual imperceptibility in practice (see Section VI). We suggest that, in practice, μ_θ might be assumed to satisfy certain smoothness constraints, for example k-Lipschitz continuity for some $0 < k < \infty$. Furthermore, the denoising schedule might inject random noise at every step. We leave theoretical security proofs for these generalised assumptions for future work but note that the training-free assumption assumes that the defender generally has no control over θ.

A perhaps more promising avenue for provable security is to explore embedding rules beyond Mix that preserve the native distributional properties of the decoder output. Of course, accurately characterizing those properties may itself be computationally prohibitive: one may need a number of samples exponential in the number of embedding steps d just to estimate even simple statistics. Crucially, though, this sampling barrier applies equally to defender and adversary, so it is not immediately clear which side would enjoy the practical advantage in a statistical detection game. Understanding how to tilt that balance in favor of the defender - perhaps by identifying distributional invariants that admit efficient testing - remains an important direction for future work.

VI. Experiments

We evaluate the performance of the PSyDUCK framework across various steganographic tasks, including both image and video applications. Our experiments cover pixel-based and latent-based diffusion models, with a focus on how controlled divergence affects recovery accuracy, encoding capacity, and detection rates. We first compare PSyDUCK to existing baselines to assess its effectiveness and scalability. Next, we examine its performance in video-based steganography. Finally, ablation studies are conducted to analyze the influence of key factors such as divergent step counts, model precision, and base image type.

Model Type	Stegosystem	Bytes	Acc.	Detection Rate	
				SRNet	SiaStegNet
Pixel-based	Pulsar	541.7	94.00	50.0	50.0
	DiffusionStego	8192	**98.46**	76.3	85.0
	StegaDDPM	512	88.62	50.0	50.0
	Psyduck ($d = 1$)	512	92.95	**50.0**	**50.0**
	Psyduck ($d = 2$)	512	97.47	56.3	50.0
	Psyduck ($d = 3$)	1536	96.47	57.4	55.0
	Psyduck ($d = 10$)	8192	95.63	76.3	80.0
Latent-based	DiffusionStego	32	72.38	74.6	75.0
	StegaDDPM	32	67.47	50.0	50.0
	Psyduck ($d = 1$)	96	94.90	**50.0**	**50.0**
	Psyduck ($d = 2$)	96	97.77	51.2	50.0
	Psyduck ($d = 3$)	96	**98.42**	51.5	60.9
	Psyduck ($d = 10$)	512	94.65	84.2	85.0

TABLE I: **Comparison of pixel-based and latent-based models.** A comparison of training-free steganography schemes across pixel- and latent-based approaches. Reported metrics include bytes encoded, transmission accuracy (Acc.), and detection rates (SRNet, SiaStegNet) for various methods, including our Psyduck stegosystem with varying divergence levels d.

A. Experimental Details

a) Models: In our experiments with pixel-based image steganography, we use four open-source diffusion models from the DDPM [15] paper: `celeb`, `bedroom`, `cat`, and `church`. For latent-based image steganography, we use Stable Diffusion (SD) version 2.1, conditioned on text inputs [1].

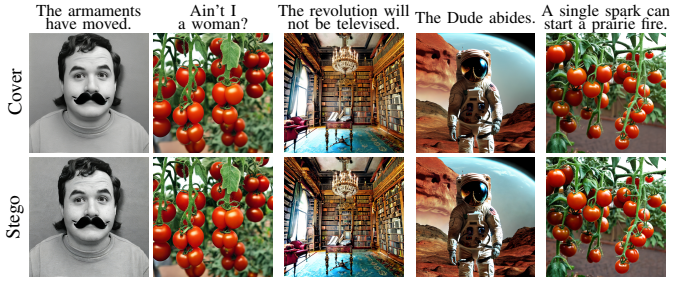
For video experiments, we use Stable Video Diffusion (SVD), an image-to-video latent model we condition with ImageNet samples.

b) Metrics: To evaluate the performance of PSyDUCK, we report the number of bytes encoded per frame, the accuracy of transmission, and the rate of detection by steganalysis tools for each of our experiments. Higher values of bytes encoded and accuracy of transmission indicate better performance, while lower detection rates indicate better performance. A 50.0% detection rate indicates that the steganalysis tool is unable to perform better than random guessing. We use two well-known steganalyzers, SRNet [16] and SiaStegNet [17].

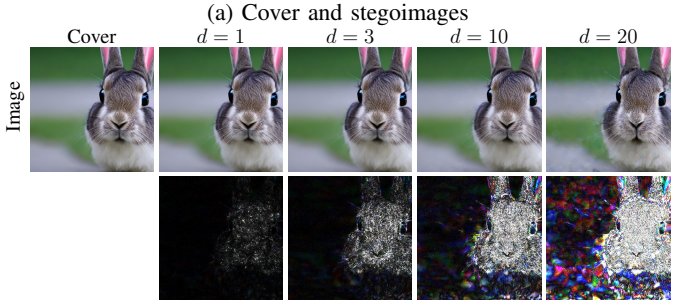
c) Data and Prompts: To generate stegosamples with SD v2.1 for latent image steganography, we use a series of prompts which we omit for brevity.

B. Image Steganography

a) Quantitative Evaluation: To ease comparisons with existing baselines, we begin by presenting experiments on established image steganography methods. Specifically, we



(a) Cover and stegoimages



(b) Effects of d

Fig. 4: **Qualitative analysis of SD v2.1 stegosamples.** a Examples of stegoimages obtained by encoding the text reported at the top on the cover image with PSyDUCK and SD v2.1. Stegoimages are perceptually undistinguishable. b Images generated from SD v2.1 using identical keys but varying divergent step count d. For easing visualization, we display the differences between cover and stego image magnified by $20\times$ in the bottom row.

evaluate against Pulsar [2], StegaDDPM [18], and Diffusion-Stego [19]. To demonstrate the versatility of PSyDUCK across different scenarios, we first compare its performance to the baselines using *pixel*-based diffusion models. Since we are encoding bitstrings, we use only two reference keys ($r = 2$) throughout the subsequent experiments. The number of bytes encoded in each model is a function of the number of divergent steps d, the number of reference keys r, and the size of the model's embedding space, whether in pixel-space for *pixel*-based models or latent-space for *latent*-based models. The results, summarized in Table I, indicate that PSyDUCK closely matches the performance of Pulsar when the number of divergent steps $d = 2$. PSyDUCK achieves slightly higher recovery accuracy but with a marginally higher detection rate by the SRNet steganalyzer. DiffusionStego, on the other hand, demonstrates significantly higher throughput and recovery accuracy, at the cost of substantially increased detection rates.

The true potential of PSyDUCK is demonstrated through experiments with *latent*-based diffusion models. We evaluate PSyDUCK performance using SD v2.1 as the model backbone and compare it against baseline methods. As shown in Table I, PSyDUCK consistently outperforms the baselines in both throughput and recovery accuracy. Specifically, for $d = 1$, PSyDUCK achieves 94.90% recovery accuracy while maintaining low detection rates of 50.0% by the SRNet steganalyzer, matching StegaDDPM's detection rates but with significantly higher throughput. Increasing the number of divergent steps to $d = 2$ further boosts recovery accuracy to 97.77%, with negligible impact on detection rates (51.2% and 50.0%). The maximum recovery accuracy of 98.42% is achieved at $d = 3$, though it comes with a slight increase in detection rates (51.5% and 60.9%). When $d = 10$, PSyDUCK sacrifices some accuracy (94.65%) in exchange for higher throughput; detection rates increase to 84.2% and 85.0%.

b) Qualitative Evaluation: We display in Figure 4a several stegoimages generated with SD v2.1 with corresponding embedded messages. As visible, the perceptual difference between cover images and stegoimages is marginal. As a further evidence of the high quality of our generated samples, we illustrate in Figure 4b the visual effect of varying the divergent step count d on images generated by SD v2.1, with differences from the cover image magnified 20 times. When $d = 1$, the differences are minimal, while more pronounced and visually significant alterations are observed as d increases, particularly at $d = 20$. Based on these findings, in subsequent experiments with latent video diffusion, we focus on divergent step counts of $d = 1$, 2, and 3.

c) Image Perceptual Metrics: We further evaluated image degradation introduced by the PSyDUCK framework using CLIPScore [20] and Fréchet Inception Distance (FID) [21]. Ideally, a good stegosystem should not introduce significant degradation into cover images. We use CLIPScore to assess semantic alignment between generated images and their textual prompts. FID quantifies perceptual differences between image distributions, hence we evaluate the FID between a collection of sampled stegoimages and their corresponding original covers. We compare using SD v2.1 with varying divergent step counts d. The CLIPScore remained relatively stable, with values of 0.8331 on the cover images, 0.8331 with $d = 1$, 0.8324 with $d = 2$, and 0.8320 with $d = 3$, indicating minimal degradation of prompt following. FID measured 0.266 for $d = 1$, 3.860 for $d = 2$, and 5.217 for $d = 3$, showing greater perceptual differences between the cover images and stegoimages as divergence increased.

C. Video Steganography

In this section, we conduct experiments with the latent video diffusion model SVD. We compare the performance of the PSyDUCK framework with the work of [22], which uses a *trained* encoder-decoder paradigm to encode messages into the latent space of a generated video.

As shown in Table II, PSyDUCK vastly outperforms the deep steganographic method proposed by [22], particularly with respect to encoding capacity. PSyDUCK encodes nearly $14\times$ more information per frame without experiencing significant drops in transmission accuracy. Even at higher divergent step counts ($d = 2$ and $d = 3$), PSyDUCK maintains robust accuracy, with minimal degradation from 96.23% at $d = 1$ to 97.95% at $d = 3$. Importantly, PSyDUCK achieves these gains while maintaining low detection rates. In contrast, [22]

Stegosystem	Bytes Encoded Per Frame	Transmission Accuracy	Detection Rate	
			SRNet	SiaStegNet
[22]	2.25	**99.42**	—	—
Psyduck ($d = 1$)	32	96.23	**50.0**	**50.0**
Psyduck ($d = 2$)	96	96.21	50.0	50.7
Psyduck ($d = 3$)	96	97.95	49.8	51.0

TABLE II: **Results on latent video diffusion experiments.** Detection rates for [22] are omitted due to the lack of an open-source implementation or reported results.

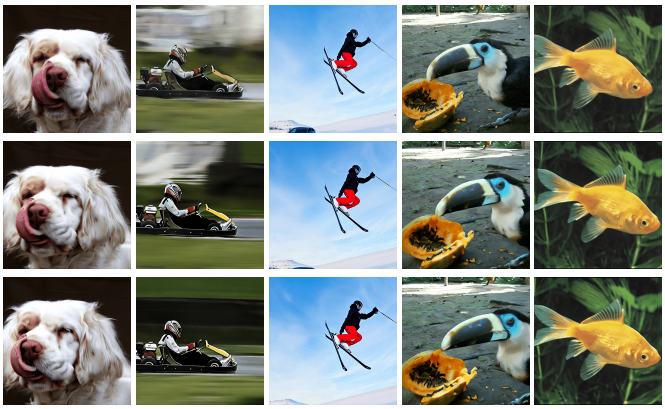


Fig. 5: **Qualitative analysis of SVD stegasamples.** We show three representative frames per video. The visual integrity of the steganographic samples remains high with no noticeable artifacts.

lacks sufficient open-source results for detection rates, but its relatively low encoding capacity suggests its effectiveness may be limited in higher-throughput settings. This demonstrates PSyDUCK's ability to balance high encoding throughput and reliable recovery accuracy with minimal susceptibility to detection, even as the divergent step count and encoded data per frame increase.

To complement our quantitative results, Figure 5 presents a qualitative analysis of the generated videos, showcasing three representative frames per video. The visual integrity of the steganographic samples remains high, with no visible artifacts that would reveal the presence of hidden information.

D. Model Precision Ablation Studies

Reducing precision from FP32 to FP16 (Table III) lowers memory and computation but incurs quantization error that degrades decoding accuracy as the divergence depth d grows (e.g. at $d = 3$, FP32 materially outperforms FP16).

VII. Discussion

We introduce **PSyDUCK**, a novel training-free steganographic framework that robustly hides information within the denoising steps of latent diffusion models. Although theoretical security guarantees remain open, PSyDUCK demonstrates efficient and visually imperceptible performance for both image and video covertexts. Future research will focus on formalizing security proofs for denoising-process based embedding, broadening our steganalysis evaluations, and extensions to in-trajectory watermarking solutions using pseudo-random codes.

Precision	Divergent Steps d	Bytes Encoded	Transmission Accuracy
fp32	1	512	75.00
	2	512	79.64
	3	**512**	**85.10**
fp16	1	512	74.21
	2	512	75.56
	3	512	80.47

TABLE III: **Effect of model precision on recovery with** `celeb`. The table highlights how `fp32` consistently outperforms `fp16`. This suggests that higher precision is critical to accurate recovery, particularly in scenarios with more divergent steps.

Acknowledgment

OpenAI ChatGPT-4.1 was used for light editing. This publication was supported by the Department for Science, Innovation and Technology and the Royal Academy of Engineering under the Research Fellowships scheme, as well as the IC Postdoctoral Research Fellowship scheme, and a generous Open Philanthropy grant.

References

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2022. [Online]. Available: https://arxiv.org/abs/2112.10752

[2] T. M. Jois, G. Beck, and G. Kaptchuk, "Pulsar: Secure steganography through diffusion models," Cryptology ePrint Archive, Paper 2023/1758, 2023. [Online]. Available: https://eprint.iacr.org/2023/1758

[3] S. Gunn, X. Zhao, and D. Song, "An Undetectable Watermark for Generative Image Models." ICLR, 2025.

[4] L. A. Bauer, W. Bao, and V. Bindschaedler, "Provably Secure Covert Messaging Using Image-Based Diffusion Processes." SatML, 2025.

[5] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," *Proceedings of 1st International Conference on Image Processing*, 1994.

[6] S. Baluja, "Hiding images in plain sight: Deep steganography," vol. 30. NeurIPS, 2017.

[7] V. Holub and J. Fridrich, "Low-complexity features for jpeg steganalysis using undecimated dct," *TIFS*, vol. 10, pp. 219–228, 02 2015.

[8] C. Schroeder de Witt, S. Sokota, J. Z. Kolter, J. Foerster, and M. Strohmeier, "Perfectly secure steganography using minimum entropy coupling." ICLR, 2023.

[9] P. Wei, Q. Zhou, Z. Wang, Z. Qian, X. Zhang, and S. Li, "Generative steganography diffusion," 2023. [Online]. Available: https://arxiv.org/abs/2305.03472

[10] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," 2015. [Online]. Available: https://arxiv.org/abs/1503.03585

[11] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," 2021. [Online]. Available: https://arxiv.org/abs/2105.05233

[12] N. J. Hopper, J. Langford, and L. von Ahn, "Provably Secure Steganography," in *Advances in Cryptology — CRYPTO 2002*, 2002.

[13] A. Müller, D. Lukovnikov, J. Thietke, A. Fischer, and E. Quiring, "Black-Box Forgery Attacks on Semantic Watermarks for Diffusion Models." CVPR 2025, Jun. 2025.

[14] L. von Ahn and N. J. Hopper, "Public-key steganography," in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 323–341.

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[16] L. Wu, C. Zhang, J. Liu, J. Han, J. Liu, E. Ding, and X. Bai, "Editing text in the wild," 2019. [Online]. Available: https://arxiv.org/abs/1908.03047

[17] W. You, H. Zhang, and X. Zhao, "A siamese cnn for image steganalysis," *TIFS*, vol. 16, pp. 1–1, 07 2020.

[18] Y. Peng, D. Hu, Y. Wang, K. Chen, G. Pei, and W. Zhang, "StegaDDPM: generative image steganography based on denoising diffusion probabilistic model," ser. MM '23, 2023, p. 7143–7151.

[19] D. Kim, C. Shin, J. Choi, D. Jung, and S. Yoon, "Diffusion-stego: Training-free diffusion generative steganography via message projection," 2024. [Online]. Available: https://openreview.net/forum?id=Ve9GKnDNDQ

[20] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "CLIPScore: a reference-free evaluation metric for image captioning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 7514–7528.

[21] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *NeurIPS*, 2017.

[22] X. Mao, X. Hu, W. Peng, Z. Gan, Q. Ying, Z. Qian, S. Li, and X. Zhang, "From covert hiding to visual editing: Robust generative video steganography," 2024. [Online]. Available: https://arxiv.org/abs/2401.00652