StegFormer: Rebuilding the Glory of Autoencoder-Based Steganography

Xiao Ke[†1,2], Huanqi Wu[†1,2], Wenzhong Guo[*1,2]

[1]Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing, College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China
[2]Engineering Research Center of Big Data Intelligence, Ministry of Education, Fuzhou 350116, China
kex@fzu.edu.cn, wuhuanqi135@gmail.com, guowenzhong@fzu.edu.cn

Abstract

Image hiding aims to conceal one or more secret images within a cover image of the same resolution. Due to strict capacity requirements, image hiding is commonly called large-capacity steganography. In this paper, we propose StegFormer, a novel autoencoder-based image-hiding model. StegFormer can conceal one or multiple secret images within a cover image of the same resolution while preserving the high visual quality of the stego image. In addition, to mitigate the limitations of current steganographic models in real-world scenarios, we propose a normalizing training strategy and a restrict loss to improve the reliability of the steganographic models under realistic conditions. Furthermore, we propose an efficient steganographic capacity expansion method to increase the capacity of steganography and enhance the efficiency of secret communication. Through this approach, we can increase the relative payload of StegFormer to **96 bits per pixel** without any training strategy modifications. Experiments demonstrate that our StegFormer outperforms existing state-of-the-art (SOTA) models. In the case of single-image steganography, there is an improvement of more than **3 dB** and **5 dB** in PSNR for secret/recovery image pairs and cover/stego image pairs.

Introduction

Steganography is a technique of information hiding whose primary goal is to conceal secret data in a carrier without affecting its representation and avoid arousing suspicion. As an essential part of steganography, image steganography uses images as a carrier for information hiding, which has a long history dating back to ancient Greece. Nowadays, research on digital image steganography is still proceeding. Traditional image steganography methods include using the LSB (Least Significant Bit) algorithm (Wang, Lin, and Lin 2001) to hide the information in bits into the least significant bits of pixels or using the DCT (Discrete Cosine Transform) transform (Patel and Dave 2012) to conceal the data into the frequency domain of the image. However, the capacity of those methods is small, which is far from meeting the needs of modern secret communication.

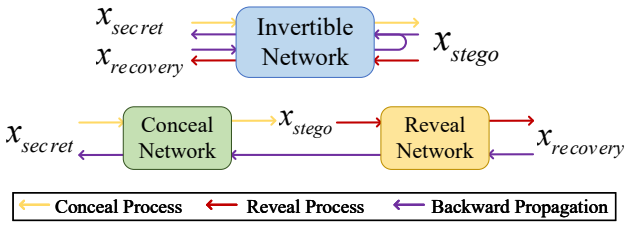

Figure 1: The illustration of difference between autoencoder based steganography and INN-based steganography.

Recently, the advancement of deep learning has led to the emergence of numerous deep steganographic models. Those models have a larger capacity than traditional methods. Consequently, image hiding, as a branch of image steganography, has seen significant development. Image hiding requires concealing one or more secret images into a carrier image, and its relative payload is beyond 24 bits per pixel (bbp). Because of stricter requirements on capacity, it is also called large-capacity steganography.

To the best of our knowledge, current large-capacity steganography frameworks can be classified into two architectures. The first is the autoencoder architecture, exemplified by BalujaNet (Baluja 2017), which consists of separate conceal network and reveal network. The second is the invertible neural network (INN) architecture, exemplified by HiNet (Jing et al. 2021), which utilizes the same parameters for hiding and recovery processes. The INN-based steganography treats the concealing and recovering processes of the secret image as a pair of inverse problems, utilizing identical parameters for both processes. This approach minimizes information loss and renders it especially suitable for large-capacity steganography.

While the INN excels in image hiding, it faces notable challenges. Firstly, during the training process, each iteration necessitates performing backward propagation twice, as illustrated in Figure 1. Consequently, INN demands double training time of autoencoder-based models with equivalent computational cost. Secondly, the invertible framework imposes several constraints on the architecture. For instance, both the conceal network and the reveal network must be entirely consistent, which presents difficulties in modifying the structure to specific requirements.

*Corresponding Author.
†These authors contributed equally.

In addition, we observe that almost all steganographic models are trained and tested under ideal experimental conditions, neglecting their performance in real-world scenarios. Current models tend to concentrate information into the high-frequency of the cover image (Zhang et al. 2020), causing some pixel values of the stego image to be invalid (out of range [0,255]). These pixel values will be truncated before propagating in transmission channels, resulting in information loss and artifacts in the recovery image.

To address the aforementioned challenges, we introduce novel solutions. Firstly, we propose StegFormer, an autoencoder-based image hiding framework. The architecture of the conceal network and reveal network in StegFormer can be individually adjusted according to specific needs, allowing for a highly flexible model design.

Secondly, we introduce a novel normalizing training strategy and a restrict loss to mitigate the model's dependence on invalid pixel values. This approach can be flexibly applied to various steganographic models with different structures.

Thirdly, we introduce an efficient steganography capacity expansion method by simply cascading multiple secret images. Through this approach, we can increase the capacity of StegFormer to **96 bits per pixel** without any training strategy modifications.

Experimental results demonstrate that StegFormer outperforms the previous state-of-the-art (SOTA) methods. For single-image hiding, StegFormer outperforms HiNet (Jing et al. 2021) by **3.1 dB** and **5.5 dB** in secret/recovery image pairs and cover/stego image pairs. StegFormer also shows the potential for multi-image hiding and outperforms DeepMIH (Guan et al. 2022) by **3.0 dB** and **3.5 dB** in secret/recovery image pairs and cover/stego image pairs.

Overall, our contributions can be summarized as follows:

- We present StegFormer, an autoencoder-based steganographic model, to challenge the dominance of INN in image hiding.
- We present a plug-and-play normalizing training strategy and a restrict loss to enhance the reliability of the steganographic models in real-world scenarios.
- We present an efficient steganography capacity expansion method, and experiments show that StegFormer establishes new SOTA on single and multi-image hiding.

Related Work

Steganography and Image Hiding

The history of steganography can be traced back to ancient Greece. According to different carriers, it can be categorized into image steganography, video steganography, and audio steganography. Traditional image steganography involves using the LSB algorithm (Wang, Lin, and Lin 2001) to conceal information in the spatial domain of the image or employing Discrete Cosine Transform (Wang, Lin, and Lin 2001) to hide data in the frequency domain of the image. However, they can only hide a small amount of data.

Recently, the advancement of deep learning has led to the emergence of numerous deep steganography techniques. Baluja (Baluja 2017) was the first to use CNN for image steganography. Subsequently, Zhu *et al.* (Zhu et al. 2018), Tancik *et al.* (Tancik, Mildenhall, and Ng 2020) and Fang *et al.* (Fang et al. 2022) used a noise layer to approximate the distortions in realistic conditions. Zhang *et al.* (Zhang et al. 2023) utilized the information at the frequency level. Luo *et al.* (Luo et al. 2020) used GAN to optimize the perceptual quality of steganographic images. Wang *et al.* (Wang, Wu, and Wang 2023) proposed the Adapter to adjust the encoding strength according to the cover image. However, these methods prioritize robustness over capacity. For instance, StegaStamp (Tancik, Mildenhall, and Ng 2020) has a relative payload of approximately 0.00125 bbp.

Image hiding is an essential branch of image steganography, which is dedicated to concealing one or multiple images within a carrier image. Due to the strict requirement on capacity, it is also called large-capacity steganography. Baluja (Baluja 2017) first applied CNN to image hiding and successfully concealed a color image into a carrier image of the same resolution. The autoencoder architecture used by BalujaNet (Baluja 2017) has been widely used in subsequent work (Ahmadi et al. 2020; Chen et al. 2023).

However, autoencoder-based steganography tends to perform poorly in balancing the quality of stego images and recovery images, while INN-based steganography does better. HiNet (Jing et al. 2021) was the the first to employ INN for image hiding and it concealed images in the wavelet domain. ISN (Lu et al. 2021) and DeepMIH (Guan et al. 2022) introduced different multi-image steganography strategies. Xu *et al.* (Xu et al. 2022) enhanced the robustness of INN by introducing a distortion model. Lan *et al.* (Lan et al. 2023) significantly improved the security of INN by embedding information into the DCT coefficients of the cover image.

While the INN excels in image hiding, it faces notable challenges in training time and architectural flexibility. Our StegFormer demonstrates that autoencoder-based steganography has great potential in image hiding.

Vision Transformer

Transformer (Vaswani et al. 2017) shows great performance in natural language processing. Inspired by Transformer's success, Dosovitskiy *et al.* proposed ViT (Dosovitskiy et al. 2020), which directly applies a pure Transformer-based architecture on the 16×16 flattened patches. However, ViT has two significant drawbacks. Firstly, the coarse patch embedding hinders ViT from capturing details (Park and Kim 2022). Secondly, the quadratic computational cost of self-attention limits its applicability to specific visual tasks.

To address the first problem, PVT (Wang et al. 2021) introduced a pyramid structure. CrossViT (Chen, Fan, and Panda 2021) adopted a dual-branch structure with different patch sizes to learn multi-scale information. Uformer (Wang et al. 2022) formed a U-shaped structure. To solve the second problem, Swin Transformer (Liu et al. 2021) introduced window-based self-attention. DAT (Xia et al. 2022) used deformable attention to avoid excessive attention computation.

Inspired by Uformer (Wang et al. 2022), we propose StegFormer, a model specifically designed for image hiding. Our experiments prove that StegFormer outperforms the previous state-of-the-art methods.

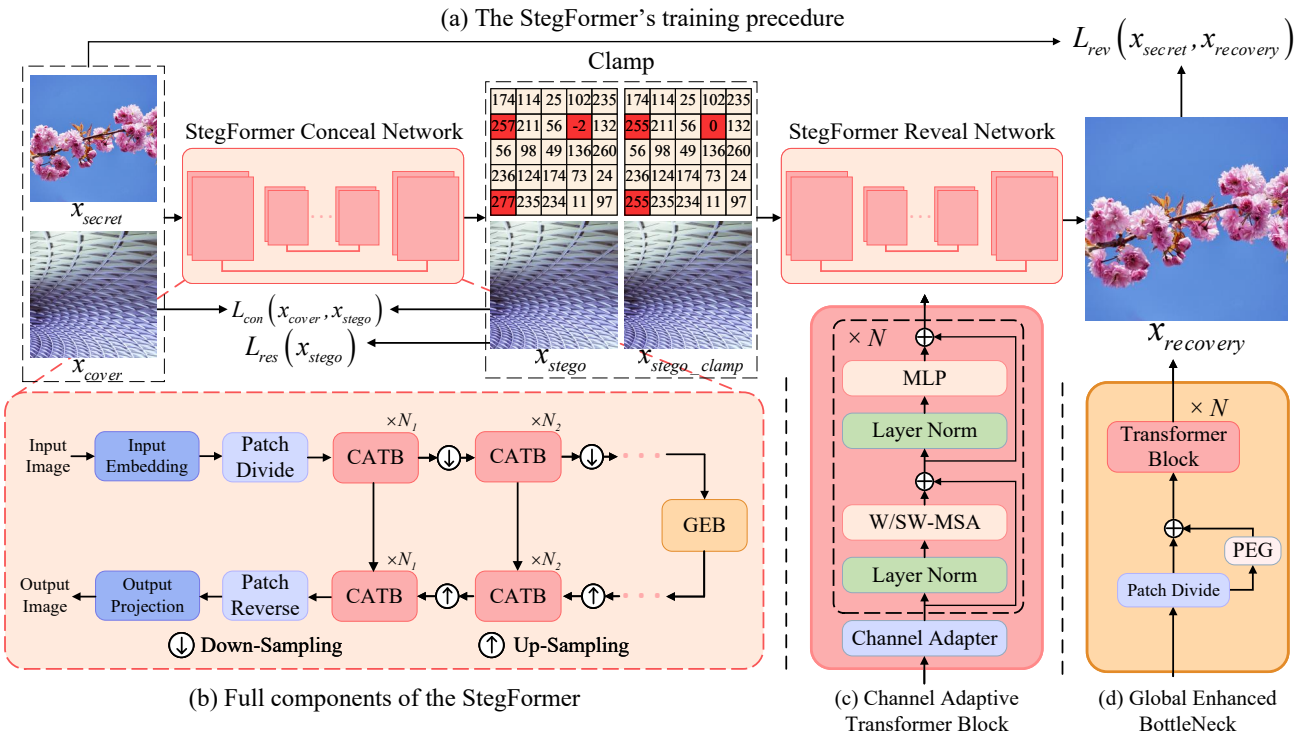

Figure 2: The framework of StegFormer. The secret image concatenates with cover image as the input of the StegFormer conceal network to generate a stego image. Then the pixel values of the stego image will be truncated to range [0,255] and input to StegFormer reveal network to recover secret image.

Proposed Method

Overall Pipeline

As shown in Figure 2 (b), the overall structure of the StegFormer is a U-shaped network with skip-connections between the encoder and decoder following the U-Net (Ronneberger, Fischer, and Brox 2015). Take StegFormer conceal network as an example. Suppose the input is a cover image $I_{cover} \in \mathbb{R}^{3\times H\times W}$ and a secret image $I_{secret} \in \mathbb{R}^{3\times H\times W}$. Then we concatenate I_{cover} and I_{secret} together to be $I_{input} \in \mathbb{R}^{6\times H\times W}$ and applies a 3×3 convolutional layer to extract low-level features $X^0_{img} \in \mathbb{R}^{C\times H\times W}$. Next, we split X^0_{img} into non-overlapping patches sized $P \times P$ by a patch divide module to get $X^0_{token} \in \mathbb{R}^{\frac{HW}{P^2}\times P^2C}$. Each patch is treated as a *token* and $\frac{HW}{P^2}$ is the number of tokens, P^2C is the dimension of each token. Following the design of the U-Net, X^0_{token} will be passed through K encoder stages. Each stage contains a stack of the proposed **C**hannel **A**daptive **T**ransformer **B**lock (CATB) and one down-sampling layer. In the down-sample layer, we use 4×4 convolutional layer with stride 2 for down-sampling, double the channels and reduce half of the resolution of the feature maps.

Then, we use **G**lobal **E**nhance **B**ottleneck (GEB) as the bottleneck stage. With the help of global self-attention, GEB can capture longer dependencies. GEB uses **C**onditional **P**ositional **E**ncoding (CPE) (Chu et al. 2021) for position embedding, which can change along with the input size and keep translation equivalence. Thanks to the GEB, StegFormer can generalize to arbitrary input resolution.

Next, we use the U-Net decoder to reconstruct features. Similar to the encoder, the decoder also consists of K stages. Each stage contains an up-sampling layer and a stack of CATB. We use 2×2 transposed convolution with stride 2 to up-sample, reduce half of the feature channels and double the resolution of the feature maps. Then the up-sampled features are concatenated with the corresponding features from the U-Net encoder through skip-connection and input to the CATB to restore the image. After K decoder stages, we get $X^{output}_{token} \in \mathbb{R}^{\frac{HW}{P^2}\times 2P^2C}$. We reshape it to 2D feature maps $X^{output}_{img} \in \mathbb{R}^{2C\times H\times W}$ and apply a 3×3 convolution to obtain stego image $X^{stego} \in \mathbb{R}^{3\times H\times W}$.

Before inputting X^{stego} to the reveal network, the pixel values of the X^{stego} will be truncated to range [0,255]. The structure of the reveal network is similar to conceal network, except that the input is stego image $X^{stego} \in \mathbb{R}^{3\times H\times W}$. After processing by the reveal network, the output is the recovery image. We train StegFormer using three different loss functions, which will be explained in the following sections.

Channel Adaptive Transformer Block

There are three main challenges to applying vision transformers for image hiding. Firstly, the coarse patch embedding mechanism leads to losing local features. Previous

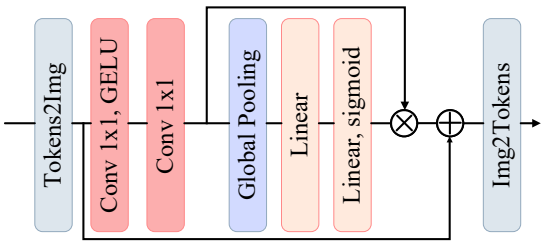

Figure 3: The illustration of the Channel Adapter.

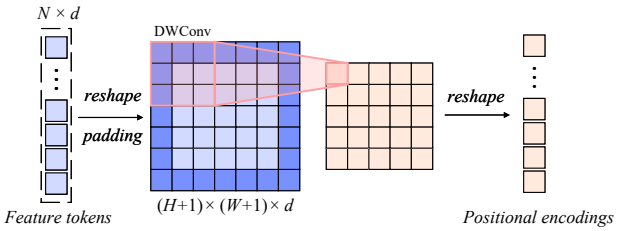

Figure 4: The illustration of Positional Encoding Generator.

work (Park and Kim 2022) has demonstrated that ViT has difficulty capturing local features which is crucial for low-level tasks like image hiding. Secondly, the global self-attention mechanism has a quadratic computational cost, making it unsuitable when input image resolution is high. Thirdly, the self-attention mechanism only considers spatial information and neglects channel information.

To address the issues mentioned above, we propose the **C**hannel **A**daptive **T**ransformer **B**lock (CATB) as the fundamental block of the StegFormer. As illustrated in Figure 2(c), CATB comprises two core modules. The first is the Channel Adapter, which utilizes the global information of each channel to adjust the distinctions between channels dynamically. The second is the Swin Transformer Block (Liu et al. 2021), which employs the window-based attention mechanism to model local features.

Channel Adapter. Channel information plays a crucial role in image hiding (Li et al. 2023). In CATB, the window-based self-attention only focuses on spatial information and neglects the channel information. The introduction of channel attention is essential to enhance the performance of the StegFormer. Inspired by SENet (Hu et al. 2017), we propose the Channel Adapter.

As shown in Figure 3, given input $X_{token} \in \mathbb{R}^{\frac{HW}{P^2} \times P^2 C}$, we first reshape the tokens to 2D feature maps $X_{img} \in \mathbb{R}^{C \times H \times W}$, then use two 1×1 convolution with a GELU non-linearity in between to capture channel information $X_{channel} \in \mathbb{R}^{C \times H \times W}$. Subsequently, we use global pooling and 2-layer MLP with Sigmoid activation to get channel attention weights $X_{weight} \in \mathbb{R}^{C}$. After that, we multiply $X_{channel}$ and X_{weight} in channels to get channel bias $X_{bias} \in \mathbb{R}^{C \times H \times W}$. Finally, we add X_{bias} to the X_{img} and reshape it into X'_{token} as the output of the Channel Adapter. Channel Adapter adjusts the strength of different channels in the form of bias to enhance the performance of the StegFormer.

Swin Transformer Block. We use Swin Transformer Block (STB) (Liu et al. 2021) to capture local features. As shown in Figure 2(c), STB consists of a window-based multi-head self-attention (W-MSA) followed by a 2-layer MLP with a GELU non-linearity in between. The W-MSA module lacks connections across windows. Therefore, Swin Transformer (Liu et al. 2021) uses a shifted window (SW) partitioning approach that alternates between two partitioning configurations in consecutive STBs. The operation of two consecutive STBs can be expressed as:

$$\begin{aligned} \hat{X}^l &= \text{W-MSA}(\text{LN}(X^{l-1})) + X^{l-1}, \\ X^l &= \text{MLP}(\text{LN}(\hat{X}^l)) + \hat{X}^l, \\ \hat{X}^{l+1} &= \text{SW-MSA}(\text{LN}(X^l)) + X^l, \\ X^{l+1} &= \text{MLP}(\text{LN}(\hat{X}^{l+1})) + \hat{X}^{l+1}, \end{aligned} \tag{1}$$

where $\hat{X}^l$ and X^l denote the output features of the (S)W-MSA module and the MLP module for block l, respectively.

Global Enhance Bottleneck

StegFormer employs window-based self-attention to extract local features in the encoder stages, limiting its ability to leverage global information. Notably, the resolution of the feature map is substantially reduced due to multiple downsampling operations, resulting in a comparatively reduced computational cost for the global self-attention operation. Therefore, we use **G**lobal **E**nhance **B**ottleneck (GEB) to model global features. GEB consists of a **P**osition **E**ncoding **G**enerater (PEG) and multiple standard Transformer Blocks (Dosovitskiy et al. 2020).

Positional Encoding Generator. As shown in Figure 2(d), GEB incorporates **C**onditional **P**ositional **E**ncoding (CPE) generated by **P**ositional **E**ncoding **G**enerator (PEG) for position embedding. It has been demonstrated in prior research (Islam, Jia, and Bruce 2020; Chu et al. 2021) that convolution can implicitly capture position information through zero padding. So we adopt depth-wise convolution as PEG, as shown in Figure 4. The CPE generated by PEG can adapt to varying input resolutions while maintaining translation equivalence. Given input $X_{token} \in \mathbb{R}^{N \times d}$, we convert it into a 2D representation $X_{token} \in \mathbb{R}^{H \times W \times d}$, where $N = H \times W$. We employ a 3×3 depth-wise convolution with zero-padding to generate positional encoding $PE_{token} \in \mathbb{R}^{H \times W \times d}$. Next, we reshape it to be $PE_{token} \in \mathbb{R}^{N \times d}$ and add it to X_{token} for position embedding.

Transformer Block with Global MSA. After position embedding, we use multiple Vision Transformer Blocks (Dosovitskiy et al. 2020) to capture global features. The process is as follows:

$$\begin{aligned} \hat{X}^l &= \text{MSA}(\text{LN}(X^{l-1})) + X^{l-1}, \\ X^l &= \text{MLP}(\text{LN}(\hat{X}^l)) + \hat{X}^l, \end{aligned} \tag{2}$$

where $\hat{X}^l$ and X^l denote the output features of the MSA module and the MLP module for block l, respectively.

By introducing CPE, GEB possesses the inductive bias inherent to convolution, which ensures the robustness of StegFormer across arbitrary resolutions. Our experiments illustrate that StegFormer, trained on images sized 256×256, performs well on images as large as 1024×1024.

Loss Function

The overall loss function comprises three distinct components: the concealing loss to guarantee the concealing performance, the revealing loss to ensure the recovering performance and a novel restrict loss to enhance the reliability of the steganographic models under realistic conditions.

Concealing loss. The concealing loss is based on Charbonnier Loss (Charbonnier et al. 1994). Given the cover image I_{cover} and the stego image I_{stego}, we define concealing loss as follows, ϵ is used to prevent the loss from being 0.

$$L_{con} = \sqrt{||I_{cover} - I_{stego}||^2 + \epsilon^2}. \tag{3}$$

For image hiding, previous work generally uses L1 Loss for training. L1 Loss is faster to convergence, but may be difficult to converge to optimal due to not differentiable at zero. However, the Charbonnier loss has a consistent gradient around zero (Lai et al. 2017), which can fully exploit the potentiality of the conceal network.

Revealing loss. The revealing loss is also based on Charbonnier Loss. Given the secret image I_{secret} and the recovery image $I_{recovery}$, we define the revealing loss as follows, where ϵ is used to prevent the loss from being 0.

$$L_{rev} = \sqrt{||I_{secret} - I_{recovery}||^2 + \epsilon^2}. \tag{4}$$

Restrict loss. We propose the restrict loss, which encourages the conceal network to normalize the representation of the stego image. Given the stego image I_{stego}, we define restrict loss as follows:

$$L_{res} = \sum_{i=1}^{H}\sum_{j=1}^{W} \begin{cases} \frac{1}{2}(I_{stego}(i,j) - 1)^2, & \text{if } I_{stego}(i,j) > 1 \\ \frac{1}{2}(I_{stego}(i,j))^2, & \text{if } I_{stego}(i,j) < 0 \\ 0. & \text{otherwise} \end{cases} \tag{5}$$

Total loss function. The total loss function L_{total} is the sum of the concealing loss L_{con}, the revealing loss L_{rev} and the restrict loss L_{res}, as follows:

$$L_{total} = L_{con} + L_{rev} + L_{res}. \tag{6}$$

Normalizing Training Strategy

Existing large-capacity steganographic models are commonly trained using the concealing loss and the revealing loss mentioned above. While this approach performs well under ideal experimental conditions, a crucial issue is overlooked: the pixel value of the stego image should be bounded within [0, 255]. Current models tend to concentrate information into the high-frequency region of the cover image (Zhang et al. 2020), which causes some pixel values of the stego image to be invalid (out of range [0,255]). Consequently, these values will be truncated before propagating in transmission channels, resulting in information loss and artifacts in the recovery image, as shown in Figure 5.

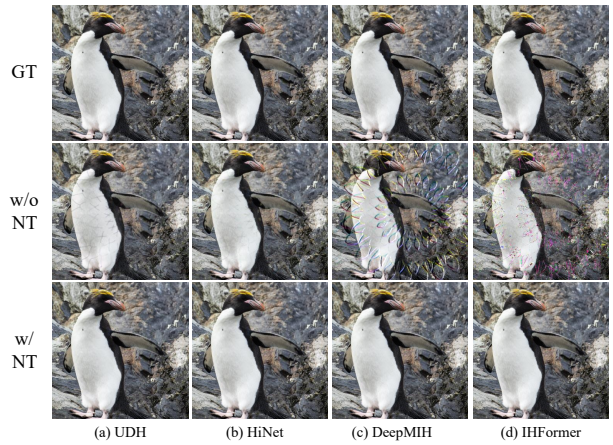

Figure 5: Visual comparisons of the recovery secret images by different methods with or without normalizing training under realistic conditions. For each image set, the first row represents the secret image. The second row represents the recovery secret image without normalizing training. The third row represents the recovery image after normalizing training. NT means normalizing training.

One straightforward solution to this problem is using a saturated activation function like Sigmoid or Tanh at the end of the conceal network (Baluja 2017). However, this approach may lead to the problem of gradient vanishing (Minai and Williams 1993), causing a substantial reduction in performance. Therefore, we propose a plug-and-play normalizing training strategy. By introducing restrict loss and performing gradient truncation, we can preserve the original performance of the steganographic model and greatly enhance its reliability in real-world scenarios. Our experiments demonstrate that this strategy applies well to other steganographic models.

Restrict loss. As explained above, we propose the restrict loss. The image will be normalized to the range of [0, 1] during the training and testing procedure, so we calculate the mean squared error (MSE) loss for the part of the stego image exceeding one and those below zero, as depicted by Equation 5.

Gradient truncation. Given stego image I_{stego}. We truncate its pixel values to the range of [0,1] to get I_{clamp} according to Equation 7. Subsequently, during the training process, we use I_{clamp} as the input of the reveal network. This approach truncates the gradients associated with invalid pixel values and prevents the conceal network from becoming reliant on these specific pixel values.

$$I_{clamp}(i,j) = \begin{cases} 1, & \text{if } I_{stego}(i,j) > 1 \\ I_{stego}(i,j), & \text{if } 0 \le I_{stego}(i,j) \le 1 \\ 0. & \text{if } I_{stego}(i,j) < 0 \end{cases} \tag{7}$$

Normalizing training. As shown in Figure 2(a), by introducing restrict loss and truncating the pixel values of the

Methods	Cover/Stego image pair											
	DIV2K (1024× 1024)				COCO (256× 256)				ImageNet (256× 256)			
	PSNR↑	SSIM↑	MAE↓	RMSE↓	PSNR↑	SSIM↑	MAE↓	RMSE↓	PSNR↑	SSIM↑	MAE↓	RMSE↓
HiDDeN	32.48	0.9172	9.73	13.93	32.98	0.9137	9.66	13.39	32.95	0.9124	9.87	13.60
UDH	44.68	0.8913	3.58	4.43	43.89	0.8988	3.79	4.65	43.87	0.9018	3.81	4.66
HiNet	50.79	0.9926	1.46	2.06	45.98	0.9806	2.43	3.56	46.00	0.9853	2.49	3.55
DeepMIH	45.73	0.9873	2.21	3.14	43.13	0.9721	2.83	3.91	43.29	0.9621	2.74	4.21
DAH-Net	49.39	0.9896	1.72	2.79	46.15	0.9856	2.96	3.84	45.75	0.9857	2.98	3.78
StegFormer (Ours)	**56.30**	**0.9956**	**0.73**	**1.23**	**48.77**	**0.9884**	**1.41**	**2.48**	**48.79**	**0.9859**	**1.51**	**2.50**
Methods	**Secret/Recovery image pair**											
	DIV2K (1024× 1024)				**COCO (256× 256)**				**ImageNet (256× 256)**			
	PSNR↑	**SSIM↑**	**MAE↓**	**RMSE↓**	**PSNR↑**	**SSIM↑**	**MAE↓**	**RMSE↓**	**PSNR↑**	**SSIM↑**	**MAE↓**	**RMSE↓**
HiDDeN	39.24	0.9502	3.54	5.29	36.29	0.9235	5.32	8.01	36.02	0.9132	5.02	7.32
UDH	42.02	0.9781	2.15	3.23	34.75	0.9175	4.82	7.79	34.62	0.9034	5.29	8.22
HiNet	52.32	0.9936	0.88	1.28	47.06	0.9796	1.82	2.78	47.07	0.9764	1.95	2.86
DeepMIH	47.92	0.9892	1.76	2.54	45.31	0.9698	2.83	3.53	45.83	0.9889	2.79	3.54
DAH-Net	50.72	0.9896	1.54	1.98	47.46	0.9726	1.54	2.17	47.35	0.9834	2.07	2.95
StegFormer (Ours)	**55.45**	**0.9964**	**0.73**	**1.08**	**49.21**	**0.9882**	**1.48**	**2.33**	**49.18**	**0.9851**	**1.62**	**2.41**

Table 1: Results on different datasets under ideal conditions, with the best results in bold and the second bests underlined.

Methods	Amount	Stego		Recovery-1		Recovery-2		Recovery-3		Recovery-4		Average-Recovery	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DeepMIH	2 images	38.76	0.9715	36.55	0.9613	37.72	0.9696	/	/	/	/	37.14	0.9655
	3 images	36.44	0.9583	32.54	0.9130	33.27	0.9251	35.81	0.9521	/	/	33.87	0.9301
	4 images	34.90	0.9432	30.34	0.8869	32.17	0.9119	33.76	0.9239	35.63	0.9479	32.98	0.9177
StegFormer(Ours)	2 images	41.82(**+3.0**)	0.9854	40.73	0.9841	40.61	0.9832	/	/	/	/	40.67(**+3.5**)	0.9836
	3 images	39.67(**+3.2**)	0.9709	36.95	0.9646	36.93	0.9642	36.81	0.9630	/	/	36.89(**+3.0**)	0.9639
	4 images	38.05(**+3.1**)	0.9224	33.37	0.9162	33.88	0.9290	33.65	0.9168	33.75	0.9175	33.66(**+0.6**)	0.9198

Table 2: Results of StegFormer and DeepMIH with different number of secret images on COCO dataset.

stego image, then training the steganographic model end-to-end, the reliability of the steganographic model in real-world conditions can be improved.

Experimental Results

Experimental Settings

Datasets and basic settings. We use DIV2K to train our StegFormer and the testing datasets comprise DIV2K (Agustsson and Timofte 2017), COCO (Lin et al. 2014), and ImageNet (Deng et al. 2009) to test the generalization ability. The AdamW optimizer is used to train StegFormer with the cosine decay strategy to decrease the learning rate to 1e-6 with the initial learning rate 1e-3. We default to conduct experiments under realistic conditions. Our code will be released in https://github.com/aoli-gei/StegFormer.

Evaluation metrics. We mainly adopt PSNR and SSIM metrics to evaluate the steganography performance. Other metrics include MAE, MSE, and RMSE. Note that we evaluate the PSNR on the Y channel in the YCbCr color space following the previous work (Jing et al. 2021).

Benchmarks. We compare StegFormer with several state-of-the-art image hiding methods, including HiDDeN (Zhu et al. 2018), UDH (Zhang et al. 2020), HiNet (Jing et al. 2021), DeepMIH (Guan et al. 2022) and DAH-Net (Zhang et al. 2023). We slightly modify HiDDeN for image hiding.

Single-image Steganography

Ideal condition. Table 1 compares the numerical results of our StegFormer with other methods under ideal conditions. To be specific, for cover/stego image pairs, our StegFormer achieves substantial PSNR improvements of **5.51 dB**, **2.62 dB**, and **2.79 dB** compared to the second-best results on the DIV2K, COCO, and ImageNet datasets, respectively. Similarly, for secret/recovery image pairs, our method exhibits significant PSNR improvements of **3.13 dB**, **1.75 dB**, and **1.83 dB** over the second-best alternatives on the DIV2K, COCO, and ImageNet datasets, respectively.

Real-world condition. In this experiment, we truncate the pixel value of the stego image to the range of [0,1]. When employing the conventional training strategy, as illustrated in Table 3, the quality of recovery image for all models notably diminish. As shown in Figure 5, evident artifacts appeared in the recovery image. During normalizing training, since we utilize the restrict loss to normalize the stego image generated by the conceal network, the quality of the stego image is slightly reduced. However, Table 3 shows the quality of the recovery image is significantly improved and the performance of StegFormer is still the best.

Multi-image Steganography

By simply cascading multiple secret images, StegFormer can be used to multi-image hiding. Table 2 shows that StegFormer outperforms DeepMIH in multi-image hiding.

Methods	NT	Cover/Stego image pair											
		DIV2K (1024× 1024)				COCO (256× 256)				ImageNet (256× 256)			
		PSNR↑	SSIM↑	MAE↓	RMSE↓	PSNR↑	SSIM↑	MAE↓	RMSE↓	PSNR↑	SSIM↑	MAE↓	RMSE↓
HiDDeN	×	32.48	0.9172	9.73	13.93	32.98	0.9137	9.66	13.39	32.95	0.9124	9.87	13.60
	✓	32.25	0.9192	9.98	14.01	32.78	0.9112	9.85	13.51	32.65	0.9104	9.97	13.78
UDH	×	44.68	0.8913	3.58	4.43	43.89	0.8988	3.79	4.65	43.87	0.9018	3.81	4.66
	✓	44.42	0.8914	3.65	4.45	43.78	0.9019	3.82	4.63	43.79	0.9025	3.84	4.64
HiNet	×	50.79	0.9926	1.46	2.06	45.98	0.9806	2.43	3.56	46.00	0.9853	2.49	3.55
	✓	49.64	0.9692	2.47	2.36	45.75	0.9582	2.60	3.69	45.95	0.9528	2.53	3.63
DeepMIH	×	45.72	0.9875	1.94	2.81	43.10	0.9821	2.81	3.88	43.31	0.9602	2.71	4.12
	✓	44.48	0.9829	2.03	2.83	42.32	0.9813	2.88	4.01	42.21	0.9810	2.90	4.23
DAH-Net	×	49.39	0.9896	1.72	2.79	46.15	0.9856	2.96	3.84	45.75	0.9857	2.98	3.78
	✓	48.78	0.9856	1.96	2.93	46.03	0.9815	3.15	3.97	45.34	0.9888	3.02	3.94
StegFormer (Ours)	×	56.30	0.9956	0.73	1.23	48.77	0.9884	1.41	2.48	48.79	0.9859	1.51	2.50
	✓	55.98	0.9928	0.81	1.28	48.70	0.9863	1.56	2.61	49.14	0.9846	1.63	2.57
Methods	**NT**	**Secret/Recovery image pair**											
		DIV2K (1024× 1024)				**COCO (256× 256)**				**ImageNet (256× 256)**			
		PSNR↑	**SSIM↑**	**MAE↓**	**RMSE↓**	**PSNR↑**	**SSIM↑**	**MAE↓**	**RMSE↓**	**PSNR↑**	**SSIM↑**	**MAE↓**	**RMSE↓**
HiDDeN	×	35.73	0.9324	4.21	7.28	33.13	0.9102	6.04	8.23	33.67	0.9194	6.28	8.98
	✓	38.93	0.9408	3.84	5.58	35.85	0.9145	5.15	7.63	35.79	0.9104	5.31	7.75
UDH	×	37.94	0.9649	3.18	5.57	32.52	0.8974	6.03	9.79	32.53	0.8843	6.60	10.28
	✓	40.99	0.9731	2.36	3.55	33.81	0.9051	5.27	8.33	33.56	0.8918	5.70	8.73
HiNet	×	44.35	0.9797	1.73	3.27	39.10	0.9561	3.00	5.41	39.10	0.9473	3.36	5.78
	✓	51.28	0.9886	1.01	1.38	45.03	0.9718	2.23	3.22	45.84	0.9764	2.34	3.12
DeepMIH	×	40.19	0.9713	2.87	3.71	37.34	0.9512	3.89	6.01	38.83	0.9452	4.14	5.78
	✓	46.87	0.9898	2.12	2.89	43.78	0.9696	2.82	3.83	43.83	0.9719	3.12	3.97
DAH-Net	×	37.73	0.9324	4.21	7.28	35.13	0.9102	6.04	8.23	35.67	0.9194	6.28	8.98
	✓	49.89	0.9796	1.82	2.38	44.92	0.9626	1.98	2.87	45.32	0.9764	2.57	3.18
StegFormer (Ours)	×	37.89	0.8976	3.43	9.17	33.59	0.8803	5.81	14.35	34.23	0.8789	5.95	13.9
	✓	53.43	0.9940	0.81	1.28	46.10	0.9829	1.79	2.86	45.98	0.9759	1.91	2.96

Table 3: Results on different datasets under realistic conditions. NT means normalizing training.

Tr. Loss		Cover/Stego image pair				Secret/Recovery image pair			
		PSNR	SSIM	MAE	RMSE	PSNR	SSIM	MAE	RMSE
×	✓	**56.54**	**0.9991**	**0.64**	**0.96**	42.26	0.9670	1.92	4.57
✓	×	53.56	0.9917	0.74	1.11	52.03	0.9917	1.25	1.86
✓	✓	55.98	0.9928	0.81	1.28	**53.43**	**0.9940**	**0.81**	**1.28**

Table 4: Ablation study on normalizing training strategy.

CA	Cover/Stego image pair				Secret/Recovery image pair			
	PSNR	SSIM	MAE	RMSE	PSNR	SSIM	MAE	RMSE
×	53.94	**0.9928**	0.95	1.78	50.89	0.9847	1.13	1.74
✓	**55.98**	**0.9928**	**0.81**	**1.28**	**53.43**	**0.9940**	**0.81**	**1.28**

Table 5: Ablation study on Channel Adapter.

Ablation Study

Effectiveness of normalizing training. Normalizing training strategy includes restrict loss and gradient truncation. As shown in Table 4, in the training of StegFormer, restrict loss and gradient truncation need to be used together to balance the quality of the stego image and recovery image.

Effectiveness of Channel Adapter. We remove the **C**hannel **A**dapter (CA) in CATB. As shown in Table 5, we can see that CA is very important to StegFormer. With the help of CA, the cover/stego and secret/recovery pairs of StegFormer improved by **2.04 dB** and **2.54 dB** in PSNR respectively. This may be because the CA introduces channel information to StegFormer, which is highly related to the way of image hiding (Zhang et al. 2020).

Conclusion

In this paper, we propose a novel image steganography model named StegFormer. StegFormer is based on autoencoder architecture, which can flexibly adjust for different scenarios and has higher flexibility than INN-based steganography. In subsequent quantitative experiments, we proved that StegFormer has stronger performance than other SOTA models, and its steganographic capacity can be extended to multiple images without any training strategy modifications. To mitigate the limitations of current steganographic models in real-world scenarios, we propose a normalizing training strategy and a restrict loss to improve the reliability of the steganographic models under realistic conditions. Experiments demonstrate that our StegFormer outperforms existing SOTA models. However, the cause of the problem current models face under real-world scenarios remains to be further explored in follow-up research.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61972097 and U21A20472, in part by the National Key Research and Development Plan of China under Grant 2021YFB3600503, in part by the Natural Science Foundation of Fujian Province under Grant 2021J01612 and 2020J01494, in part by the Major Science and Technology Project of Fujian Province under Grant 2021HZ022007.

References

Agustsson, E.; and Timofte, R. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 126–135.

Ahmadi, M.; Norouzi, A.; Karimi, N.; Samavi, S.; and Emami, A. 2020. ReDMark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 146: 113157.

Baluja, S. 2017. Hiding images in plain sight: Deep steganography. *Advances in Neural Information Processing Systems*, 30.

Charbonnier, P.; Blanc-Féraud, L.; Aubert, G.; and Barlaud, M. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. *Proceedings of 1st International Conference on Image Processing*, 2: 168–172 vol.2.

Chen, C.-F. R.; Fan, Q.; and Panda, R. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 357–366.

Chen, H.; Zhu, T.; Zhao, Y.; Liu, B.; Yu, X.; and Zhou, W. 2023. Low-frequency Image Deep Steganography: Manipulate the Frequency Distribution to Hide Secrets with Tenacious Robustness. *arXiv preprint arXiv:2303.13713*.

Chu, X.; Tian, Z.; Zhang, B.; Wang, X.; Wei, X.; Xia, H.; and Shen, C. 2021. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fang, H.; Jia, Z.; Ma, Z.; Chang, E.-C.; and Zhang, W. 2022. PIMoG: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2267–2275.

Guan, Z.; Jing, J.; Deng, X.; Xu, M.; Jiang, L.; Zhang, Z.; and Li, Y. 2022. DeepMIH: Deep invertible network for multiple image hiding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 372–390.

Hu, J.; Shen, L.; Albanie, S.; Sun, G.; and Wu, E. 2017. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42: 2011–2023.

Islam, M. A.; Jia, S.; and Bruce, N. D. 2020. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*.

Jing, J.; Deng, X.; Xu, M.; Wang, J.; and Guan, Z. 2021. HiNet: deep image hiding by invertible network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4733–4742.

Lai, W.-S.; Huang, J.-B.; Ahuja, N.; and Yang, M.-H. 2017. Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41: 2599–2613.

Lan, Y.; Shang, F.; Yang, J.; Kang, X.; and Li, E. 2023. Robust Image Steganography: Hiding Messages in Frequency Coefficients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 14955–14963.

Li, H.; Liu, H.; Guo, S.; Zhou, M.; Wang, N.; Xiang, T.; and Zhang, T. 2023. Smaller Is Bigger: Rethinking the Embedding Rate of Deep Hiding. *ArXiv*, abs/2302.11918.

Lin, T.-Y.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European conference on computer vision (ECCV)*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Lu, S.-P.; Wang, R.; Zhong, T.; and Rosin, P. L. 2021. Large-capacity image steganography based on invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10816–10825.

Luo, X.; Zhan, R.; Chang, H.; Yang, F.; and Milanfar, P. 2020. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13548–13557.

Minai, A. A.; and Williams, R. D. 1993. On the derivatives of the sigmoid. *Neural Networks*, 6(6): 845–853.

Park, N.; and Kim, S. 2022. How do vision transformers work? *arXiv preprint arXiv:2202.06709*.

Patel, H.; and Dave, P. 2012. Steganography technique based on DCT coefficients. *International Journal of Engineering Research and Applications*, 2(1): 713–717.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241. Springer.

Tancik, M.; Mildenhall, B.; and Ng, R. 2020. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2117–2126.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wang, B.; Wu, Y.; and Wang, G. 2023. Adaptor: Improving the Robustness and Imperceptibility of Watermarking by the Adaptive Strength Factor. *IEEE Transactions on Circuits and Systems for Video Technology*.

Wang, R.-Z.; Lin, C.-F.; and Lin, J.-C. 2001. Image hiding by optimal LSB substitution and genetic algorithm. *Pattern recognition*, 34(3): 671–683.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 568–578.

Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; and Li, H. 2022. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17683–17693.

Xia, Z.; Pan, X.; Song, S.; Li, L. E.; and Huang, G. 2022. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4794–4803.

Xu, Y.; Mou, C.; Hu, Y.; Xie, J.; and Zhang, J. 2022. Robust invertible image steganography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7875–7884.

Zhang, C.; Benz, P.; Karjauv, A.; Sun, G.; and Kweon, I. S. 2020. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Advances in Neural Information Processing Systems*, 33: 10223–10234.

Zhang, L.; Lu, Y.; Li, J.; Chen, F.; Lu, G.; and Zhang, D. 2023. Deep adaptive hiding network for image hiding using attentive frequency extraction and gradual depth extraction. *Neural Computing and Applications*, 1–19.

Zhu, J.; Kaplan, R.; Johnson, J.; and Fei-Fei, L. 2018. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, 657–672.