Proposal of a Random Encoding Layer Compatible with Arbitrary Message Lengths for DiffuseTrace

Ou Egami and Masaki Kawamura†
Graduate School of Sciences and Technology for Innovation, Yamaguchi University
† E-mail: m.kawamura@m.ieice.org Tel: +81-83-933-5701

***Abstract*—We propose an improved model for DiffuseTrace, which is a watermarking method that uses latent diffusion models. First, DiffuseTrace encodes a message into a watermark. Then, it inputs the watermark into a variational autoencoder (VAE). The resulting latent space is then used to generate the stego-image. However, this method of encoding messages has one limitation: the length of the message is fixed, requiring VAE reconstruction whenever the length changes. To solve this problem, we propose a converter consisting of an encoder and a decoder based on random networks. Our method enables the handling of arbitrary message lengths without reconstructing the VAE. We evaluated the image quality of stego-images generated using this method, as well as the accuracy and robustness of the watermarks against attacks. The results confirmed that our method achieves higher accuracy and robustness against attacks while providing image quality comparable to that of DiffuseTrace. Furthermore, we found that our method can support message lengths up to 20 times longer than DiffuseTrace.**

I. Introduction

Image generation artificial intelligence (AI) is a promising technology since it enables users to easily create high-quality images by describing the desired subject in a prompt. Examples of these AIs include DALL-E [1], Imagen [2], and Stable Diffusion [3]. However, these models can also be used to create realistic fake images for malicious purposes. Therefore, there is a need for ethical and legal discussions regarding the use of image-generating AI [4]. One promising type of generative model used for image generation is the diffusion model [5], [6]. These models consist of a forward diffusion process that adds noise to an image and an inverse diffusion process that removes the noise using a deep learning model. Diffusion models offer high generative quality, a variety of applications, and remarkable learning stability, which was not found in previous generative models [3]. However, they tend to be computationally expensive because they process images at their original size. Consequently, the latent diffusion model (LDM) [3] was proposed as a model that overcomes the shortcomings of the diffusion model. LDMs incorporate a variational autoencoder (VAE) [7] into diffusion models to achieve high-quality and efficient image generation. A VAE is a network that maps input data into a low-dimensional latent space and reconstructs the original data from that space. Using the latent space generated by the VAE significantly reduces the computational cost of the LDM while maintaining high image generation capability. During image generation, a trained LDM can produce images based on a prompt using only random initial noise.

As AI technology for generating images advances, it becomes increasingly difficult to distinguish between generated and real images. There is a risk of misinformation spreading due to the misuse of these images [8]. Therefore, it is necessary to explicitly indicate when and by whom the image was generated. One technique for this purpose is watermarking, which embeds invisible information into images. Watermarking embeds information by making imperceptible changes to digital content [9]. Using the creator's information as a watermark can identify the creator and also distinguish the generated image from the real one. However, Zhao *et al.* [10] recently demonstrated that watermarks embedded using different post-processing techniques [11] can be effectively removed through image reconstruction with diffusion models.Consequently, methods for embedding watermarks during image generation are necessary.

DiffuseTrace [12] is a watermarking method that can resist removal attacks and achieve high extraction accuracy, image quality, and robustness. It embeds the watermark into the initial noise of a LDM. A dedicated VAE is introduced for embedding and extracting the watermark. This VAE, referred to as the watermarking VAE, is separate from the VAE inherent in the LDM. DiffuseTrace is a multi-bit watermarking method that can embed any N-bit message. Before inputting an arbitrary message into the watermarking VAE, the encoder network encodes it into a $64 \times 64 \times N$ dimensional watermark. Since the dimension of the input layer of the VAE depends on N, when the message length N changes, the VAE needs to be reconstructed. To address this, we introduce a random network with a constant output layer dimension as the message encoder. We have made two contributions. First, the proposed model can embed messages of any length without reconfiguring the VAE. Second, the model is both scalable and flexible.

The rest of this paper is organized as follows: Section II provides detailed explanations of the Variational Autoencoder and the Latent Diffusion Model architectures. Section III introduces DiffuseTrace, an existing watermarking method relevant to this study. Section IV describes the proposed method. Section V presents the experimental setup and discusses the evaluation results. Finally, Section VI concludes the paper.

II. Background

A. Variational Autoencoder

The VAE [7] is the main network used for both generating images inside the LDM and converting watermarks into latent

979-8-3315-7206-8/25/$31.00

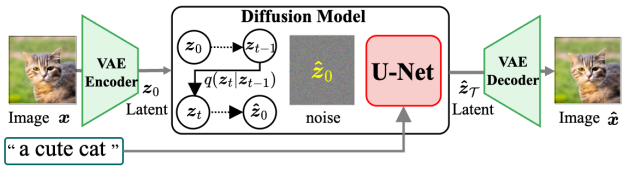

Fig. 1. Structure of Latent Diffusion Model: The latent space generated from VAE encoder is input into the diffusion model to create an image.

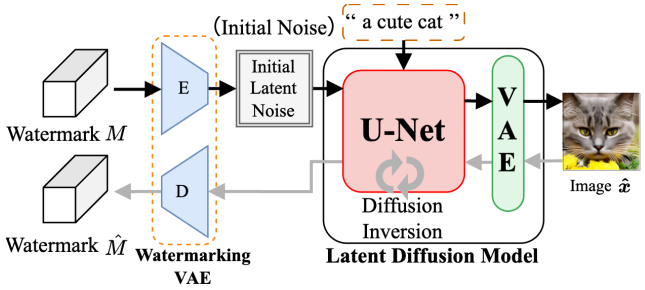

Fig. 2. Structure of DiffuseTrace: E and D represent the encoder and decoder of watermarking VAE, respectively.

variables. The VAE is a generative model composed of two structures: an encoder and a decoder. It learns an identity mapping between its inputs and outputs. Specifically, when a tensor of size $H \times W \times C$ is input to the VAE encoder, it is mapped to a lower-dimensional latent space. The VAE's latent space is represented by a high-dimensional normal distribution characterized by a mean vector $\boldsymbol{\mu}$ and a standard deviation vector $\boldsymbol{\sigma}$. Consequently, the latent variable $\boldsymbol{z}$ in the latent space is expressed as

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}. \tag{1}$$

where, $\boldsymbol{\epsilon}$ is noise that follows a standard normal distribution, $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, where $\boldsymbol{I}$ is the identity matrix. The symbol $\odot$ denotes the Hadamard product. The latent variable $\boldsymbol{z}$ is then input into the decoder, which reconstructs the original input $\boldsymbol{x}$.

When this VAE is applied to an image, the latent space is obtained as a compressed representation of the image. Conversely, an image can be generated by inputting arbitrary Gaussian noise into the image VAE's decoder. Similarly, when this VAE is applied to a watermark, the latent space provides a compressed representation of the message. The original message can be restored by inputting arbitrary Gaussian noise into the watermarking VAE's decoder.

B. Latent Diffusion Model

The LDM consists of a diffusion model and a VAE [3]. The model can generate high-quality and efficient images. Figure 1 illustrates the structure of the LDM. First, a high-dimensional image $\boldsymbol{x}$ is compressed by the VAE's encoder and mapped into a low-dimensional latent space $\boldsymbol{z}_0$. This latent space $\boldsymbol{z}_0$ is then input as the initial value of the diffusion model. The diffusion model has two processes: a forward diffusion process and an inverse diffusion process. In the forward diffusion process, each step is represented by a Markov process. At each step, Gaussian noise, represented by ϵ_t, is added to the latent space $\boldsymbol{z}_t$. Finally, the latent space is approximated by a Gaussian distribution.

The inverse diffusion process transforms noise back into the original latent space, which is necessary for generating images. However, it is difficult to formulate this inverse process. Thus, the inverse diffusion process is trained using the U-Net [13]. The trained U-Net can estimate the noise added during the diffusion process. Then, the original latent space can be reconstructed by removing the estimated noise. Finally, the latent space obtained by the inverse process is input into the VAE decoder. Then, an image $\hat{\boldsymbol{x}}$ is generated.

III. Related Work

A. DiffuseTrace

DiffuseTrace [12] is a watermarking method that uses a LDM. It consists of a LDM for image generation and a VAE for encoding and decoding messages. We call this VAE the watermarking VAE to distinguish it from the VAE inherent in the LDM. Roughly speaking, the watermark is embedded in the initial noise of the LDM when generating the stego-image. However, the *embedding* operation does not actually exist. As described in II-A, the stego-image is strictly speaking generated by providing the latent noise output from the watermarking VAE as input to the image VAE's decoder.

In DiffuseTrace, an N-bit message $\boldsymbol{m} = (m_1, m_2, \ldots, m_N)^\top$, where $m_c \in \{0, 1\}\,(c = 1, 2, \cdots, N)$, is encoded into a $64 \times 64 \times N$-dimensional tensor. This tensor is used as the watermark $\boldsymbol{M}$. The value of M_{hwc} corresponding to the c-th channel of the watermark $\boldsymbol{M}$ is given by $M_{hwc} = m_c$, where $h = 1, 2, \cdots, 64$ and $w = 1, 2, \cdots, 64$. In other words, a 64×64 matrix of zeros or ones is generated for each channel. The watermark $\boldsymbol{M}$ is encoded by the watermarking VAE's encoder. The resulting latent variable is then used as the initial value for the LDM. This latent variable is referred to as the initial latent noise. When the initial latent noise is provided, the LDM generates an image. Consequently, the generated stego-image implicitly contains the watermark.

Next, we will describe the message decoder. When the stego-image is input into the LDM, the latent noise is restored via an inverse diffusion process. The restored latent noise is then fed into the decoder of the watermarking VAE. As a result, the estimated watermark $\hat{\boldsymbol{M}}$ is obtained. By spatially averaging its component $\hat{M}_{hwc}$, the c-th element of the message is given by

$$\bar{m}_c = \frac{1}{64 \times 64} \sum_{h=1}^{64} \sum_{w=1}^{64} \hat{M}_{hwc}. \tag{2}$$

After applying thresholding to the obtained value $\bar{m}_c$, the c-th estimated message $\hat{m}_c$ becomes

$$\hat{m}_c = \begin{cases} 1 & (\bar{m}_c \geq 0.5) \\ 0 & (\bar{m}_c < 0.5) \end{cases}. \tag{3}$$

In summary, DiffuseTrace increases the redundancy of the watermark significantly by replicating the message m_c across

a 64×64 spatial dimension. The dimension of this watermark is proportional to the message length N. Consequently, the dimension of the input to the watermarking VAE is also proportional to N. Since DiffuseTrace does not require a secret key for image generation, any user can extract the watermark from the stego-image.

B. Pre-training of Watermarking VAE

Pre-training the watermarking VAE is necessary to generate latent noise that represents the watermark. To enable the VAE to embed arbitrary watermarks, it is trained to learn an identity mapping, which faithfully reconstructs the input watermark through its latent noise. The loss function is defined as the mean squared error (MSE) between the original and estimated watermarks, as given by

$$\mathrm{MSE} = \frac{1}{64 \times 64 \times C} \sum_{c=1}^{C} \sum_{h=1}^{64} \sum_{w=1}^{64} \left(M_{hwc} - \hat{M}_{hwc}\right)^2, \quad (4)$$

where C is the number of channels of the watermark, which equals the message length N. Through pretraining, the encoder learns to generate latent noise representing the watermark, and the decoder learns to reconstruct the watermark from the noise.

C. Fine-tuning Watermarking VAE

It is necessary that images can be generated from the latent noise produced by the watermarking VAE, not the image VAE. Similarly, the watermark must be recoverable from the latent noise obtained through the inverse diffusion process from a generated image. Additionally, the watermark should be recoverable even if the generated image is subject to attacks. Therefore, we fine-tune the watermarking VAE together. This improves the accuracy of watermark estimation and enhances the decoder's robustness against attacks [12].

We will explain the fine-tuning procedure in detail. First, the watermark $\boldsymbol{M}$ is input into the encoder of the watermarking VAE to generate initial latent noise. Next, the initial latent noise and a prompt are provided to the U-Net in the LDM to generate an image. Then, the generated image is subjected to random attacks, such as the addition of Gaussian noise, Gaussian blurring, and image compression. Next, the watermark is recovered from the attacked image. An inverse diffusion process is then performed on the generated image to reconstruct the latent noise. The reconstructed latent noise is then input into the decoder of the watermarking VAE to obtain the estimated watermark $\hat{M}$. The MSE (4) is used as the loss function for fine-tuning.

IV. Proposed Method

In DiffuseTrace, the watermark $\boldsymbol{M}$ is obtained by replicating each bit of the message, m_c, spatially into a 64×64 matrix. The number of channels, C, of the watermark is equal to the message length N. Thus, if the message length N changes, the watermarking VAE must be reconstructed. Furthermore, since computational complexity depends on message length, there is a limit to how long the message can be. To solve this problem, we propose a model that uses random networks to encode the message $\boldsymbol{m}$, where the number of channels C is kept constant. This means that the watermarking VAE does not need to be rebuilt.

As shown in Figure 3, the proposed model consists of encoder and decoder. Each of these networks is composed of two fully connected layers (FCLs). The weights of these layers are initialized randomly and fixed without further training. A linear function is used as the activation function for all layers. For the encoder, the weights of each FCL are initialized according to a normal distribution based on the Xavier initialization method [14]. Let W_1^0 and W_2^0 be the initial weight matrices for the input-to-hidden and hidden-to-output layers, respectively. The weight W_l^0 of the l-th layer is initialized by

$$W_l^0 \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_l^{\mathrm{in}} + n_l^{\mathrm{out}}}}\right), \; l = 1, 2, \quad (5)$$

where n_l^{in} and n_l^{out} are the numbers of input and output nodes, respectively, for the l-th layer. Since orthogonal matrices are required, we perform QR decomposition on the initial weight matrices. This results in $W_l^0 = Q_l R_l, l = 1, 2$.The resulting orthogonal matrices Q_1 and Q_2 form the final weights for the respective fully connected layers: $W_1 = Q_1$ and $W_2 = Q_2$.

The message is transformed into a watermark via the encoder. When a message $\boldsymbol{m} \in \{0, 1\}^N$ is input into the encoder with the weight $W_1 \in \mathbb{R}^{4096 \times N}$, the output of $4,096$-dimensional hidden layer is obtained as

$$\boldsymbol{\phi}_1 = W_1 \boldsymbol{m}. \quad (6)$$

Then, the output of the output layer is obtained using the weight $W_2 \in \mathbb{R}^{4096 \times 4096}$ by

$$\boldsymbol{\phi}_2 = W_2 \boldsymbol{\phi}_1 + \boldsymbol{b}, \quad (7)$$

where $\boldsymbol{b}$ is a bias. The value of each element of the bias vector $\boldsymbol{b}$ is 0.5. The watermarking VAE in DiffuseTrace receives a binary watermark as input. However, approximately half of the output of the proposed network contains negative values. Adding a positive bias reduces the number of negative values. This allows the VAE to be used as is. The output $\boldsymbol{\phi}_2$ is reshaped into a 64×64 dimensional matrix and replicated C times by the replication layer. After this process, a watermark $\boldsymbol{M} \in \mathbb{R}^{64 \times 64 \times C}$ is obtained with C channels. The number of channels, C, can be determined independently of the message length N.

The decoder performs the inverse transformation using $W_1^\top$ and $W_2^\top$, the transpose matrices of the weights W_1 and W_2 used in the encoder. When the watermark $M \in \mathbb{R}^{64 \times 64 \times C}$ is inputted into the decoder, the 64×64 matrix of the c-th channel is first reshaped into a 1-dimensional vector, $\boldsymbol{\phi}_2^c \in \mathbb{R}^{4096}$. Next, a 4096-dimensional hidden layer output is given by

$$\boldsymbol{\phi}_1^c = W_2^\top \left(\boldsymbol{\phi}_2^c - \boldsymbol{b}\right), c = 1, 2, \cdots, C. \quad (8)$$

Once the C outputs of the hidden layer, $\boldsymbol{\phi}_1^c$, are obtained, the mean vector $\bar{\boldsymbol{\phi}}_1$ is obtained by averaging them, as given by

$$\bar{\boldsymbol{\phi}}_1 = \frac{1}{C} \sum_{c=1}^{C} \boldsymbol{\phi}_1^c, \quad (9)$$

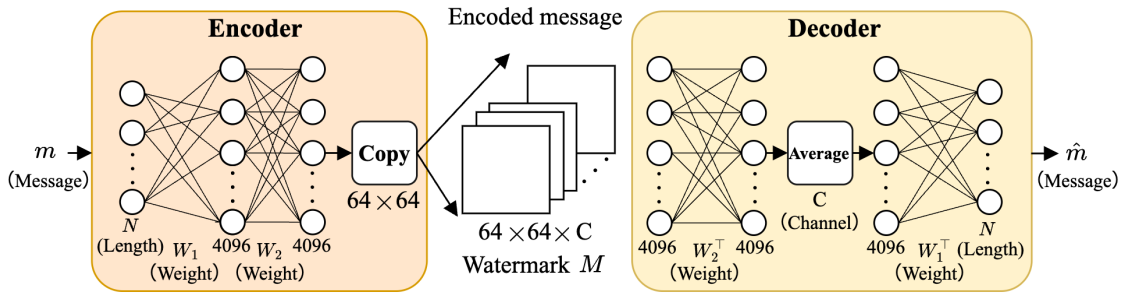

Fig. 3. Proposed random networks: The encoder converts a message into a watermark, while the decoder converts the watermark back into the original message.

Then, an N-dimensional output is obtained by

$$\tilde{\boldsymbol{m}} = W_1^\top \bar{\phi}_1. \tag{10}$$

By applying thresholding to the c-th element of the output, the c-th element of the estimated message, $\hat{m}_c$, is given by

$$\hat{m}_c = \begin{cases} 1 & (\tilde{m}_c \geq 0.5) \\ 0 & (\tilde{m}_c < 0.5) \end{cases}. \tag{11}$$

The output dimension of the proposed random network does not depend on the message length N but rather on the number of channels C, which can be chosen arbitrarily. Therefore, it is unnecessary to reconstruct the watermarking VAE if the message length changes.

V. Computer Simulation

We evaluate the quality of images generated by DiffuseTrace and our proposed model, their fidelity to corresponding prompts, and the accuracy of watermarks. Additionally, we subject the generated images to attacks to assess the accuracy and robustness of the watermarks.

A. Evaluation Metrics and Experimental Conditions

The accuracy of the estimated watermark $\hat{\boldsymbol{m}}$ obtained by the decoder of the watermarking VAE is evaluated by the accuracy (Acc) metric. For an N-bit message $\boldsymbol{m} = (m_1, m_2, \ldots, m_N)^\top$ and its estimated message $\hat{\boldsymbol{m}} = (\hat{m}_1, \hat{m}_2, \ldots, \hat{m}_N)^\top$, where $m_i, \hat{m}_i \in \{0, 1\}$, the accuracy Acc is expressed as:

$$\mathrm{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left[m_i = \hat{m}_i\right], \tag{12}$$

where the indicator function $\mathbb{1}[\cdot]$ returns 1 if the given condition holds true and 0 otherwise.

We evaluate the quality of the generated images using objective assessment metrics that require no reference images: The Natural Image Quality Evaluator (NIQE) [15] and the Perceptual Image Quality Evaluator (PIQE) [16]. NIQE assesses the naturalness of an image by calculating the discrepancy between its statistical features and those of a pre-trained statistical model of natural images. A smaller NIQE value indicates that the image being evaluated is closer to a natural image, while a larger value suggests that the image is likely artificial or distorted.

TABLE I
Watermark Accuracy, Image Quality, and Clip Score by Channel Number C ($N = 48$)

Method	C	Acc	NIQE ↓	PIQE ↓	Clip ↑
No-Watermark	–	–	4.47	26.18	0.348
DiffuseTrace	48	0.965	4.28	25.59	0.338
Proposed Method	16	0.950	4.58	27.52	0.360
	48	**0.976**	4.54	27.86	0.366
	96	**0.990**	4.52	26.62	0.362

PIQE evaluates the quality of generated images by dividing an image into small blocks and calculating quality degradation features for each block. A smaller PIQE value indicates better perceptual quality. Using these metrics, we can quantitatively evaluate the impact of watermark embedding on the naturalness and perceptual quality of generated images. Finally, we evaluate the semantic consistency between the generated images and their corresponding prompts using the Clip score [17]. A higher Clip score indicates greater semantic consistency.

The dataset used for training and evaluation was Diffusion Prompts [18]. This dataset consists of approximately 80,000 prompts. Stable Diffusion v2-1-base was used as the latent diffusion model. The watermarking VAE was trained using backpropagation, and the optimization method was Adam [19]. The learning rate was set to $\alpha = 0.0005$, and the other parameters were set to their recommended values.

B. Performance Evaluation of the Proposed Model

We evaluated using 50 randomly selected prompts from Diffusion Prompts [18]. We measured the accuracy, NIQE, PIQE, and Clip score 50 times each for images generated by DiffuseTrace and our proposed method, as well as for images with no watermark. We conducted the evaluation using the average values of these metrics. The results are shown in Table I. Our proposed method allows the number of channels C and the message length N to be selected independently. Therefore, we fixed the message length at $N = 48$ and evaluated the performance for channel numbers $C = 16, 48, 96$. Regarding watermark accuracy, our proposed method showed slightly lower accuracy than DiffuseTrace for $C = 16$ but higher accuracy for $C = 48$ and $C = 96$. These results confirm the tendency for accuracy to improve with an increased number of channels.

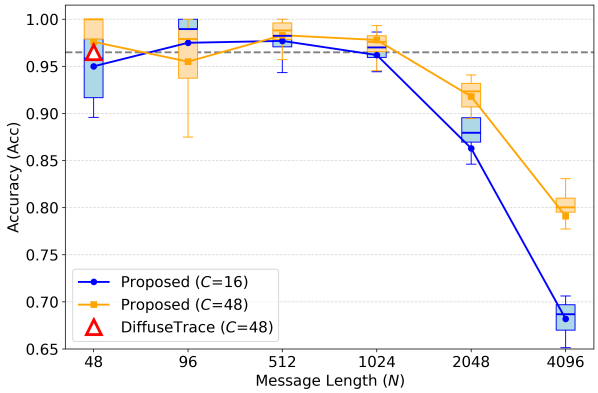

Fig. 4. Watermark accuracy (Acc) vs. message length N at $C = 16$ and $C = 48$

Next, we evaluated the quality of the generated images. Based on the NIQE and PIQE values, no significant difference was found between the generated images.NIQE and PIQE are both no-reference metrics that differ from metrics like Peak Signal-to-Noise Ratio (PSNR) [20] in their evaluation perspective. Therefore, we concluded that embedding the watermark in the LDM does not directly affect the quality of the generated images. Similarly, we confirmed that the proposed method does not significantly degrade image quality.

Finally, we evaluated the Clip scores of the generated images. The proposed method produced higher scores than DiffuseTrace. Thus, it was confirmed that the semantic consistency between the generated images and the prompts was good. Based on these results, we decided to use accuracy as the evaluation metric for subsequent evaluations. Since DiffuseTrace was only evaluated up to $C = N = 48$, subsequent evaluations will be conducted at $C = 16$ and $C = 48$.

C. Evaluation with variable message length

Our proposed method can generate stego-images using the same watermarking VAE, even when the message length N varies. Therefore, we evaluated watermark accuracy with respect to message length. For this evaluation, we set the number of channels to $C = 16, 48$, and the message lengths to $N = 48, 96, 512, 1024, 2048$ and 4096. The results are presented in Figure 4. The horizontal axis represents the message length N and the vertical axis represents the accuracy Acc. The results from 50 samples are shown as Box-and-Whisker plots. The accuracy of DiffuseTrace at $N = 48$ (Acc = 0.965) is indicated by a dashed line for comparison. No significant difference in accuracy was observed based on the number of channels when the message length was $N \leq 1024$. Furthermore, the accuracy generally exceeded 0.95, indicating sufficient performance. Conversely, for $N > 1024$, higher accuracy was observed with $C = 48$ than with $C = 16$. However, accuracy decreased as N increased. We confirmed that there is a limit to the recoverable message length.

D. Evaluation of robustness against attacks

As described in III-C, DiffuseTrace improves its robustness against attacks by applying attacks to the generated stego-images. In this section, we evaluate the accuracy of watermarks when various attacks are applied to images generated by our proposed method. Similar to the experiments mentioned above, we used 50 randomly selected prompts from Diffusion Prompts. The same attacks used in DiffuseTrace [12] are applied to the generated images, as listed below:

- ”Brightness” – Brightness adjustment (scaling factor: 2.0)
- ”Noise” – Addition of Gaussian noise (SD: 0.05)
- ”Contrast” – Contrast adjustment (scaling factor: 2.0)
- ”Hue” – Hue adjustment (scaling factor: 0.25)
- ”JPEG” – JPEG compression (quality: 50)
- ”Blur” – Gaussian blur (kernel size: 7, SD: 1.0)
- ”Resize” – Resizing (scaling factor: 0.3)
- ”BM3D” – BM3D denoising algorithm (PSNR SD: 30)

Various attacks were applied to images generated by DiffuseTrace and our proposed method. The accuracy (Acc) was measured 50 times for each method, and the average values are presented in Table II. The accuracy of the proposed method using $C = 16$ channels was slightly lower than that of DiffuseTrace. This is likely due to the reduced input dimension of the watermarking VAE. However, when using $C = 48$, the proposed method often exhibited higher accuracy than DiffuseTrace with the same number of channels across various message lengths N. Based on these results, we can conclude that the proposed method is more robust when the number of channels is equal. Additionally, the highest accuracy was achieved with a message length of $N = 512$, suggesting that our method can support longer messages than DiffuseTrace.

VI. Conclusion

DiffuseTrace [12] is a digital watermarking method that uses a VAE to embed a watermark prior to generating an image with a latent diffusion model. In DiffuseTrace, messages are redundantly encoded bit by bit and input as watermarks into the VAE. In other words, the number of watermark channels equals the message length. Consequently, the VAE had to be reconstructed whenever the message length changed. In this paper, we propose a model that uses random networks for message encoding. This model can handle arbitrary message lengths and eliminates the need for VAE reconstruction. The proposed random network consists of an encoding layer that converts messages into watermarks and a decoding layer that converts watermarks back into messages. The main advantage of the proposed method is the ability to vary the message length N using the same VAE. We achieved an accuracy greater than 0.95 when N was up to $1,024$ bits.

These evaluations show that the proposed method can adapt more easily to changes in message length. It is also more accurate and robust than DiffuseTrace.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers JP20K11973 and JP24K15106, and Support Center for

TABLE II
WATERMARK ACCURACY FROM GENERATED IMAGES UNDER ATTACK

Method (parameter)	*C*	*N*	**Brightness** 2.0	**Noise** 0.05	**Contrast** 2.0	**Hue** 0.25	**JPEG** 50	**Blur** 7×7	**Resize** 0.3	**BM3D** 30
DiffuseTrace	48		0.915	0.946	0.941	0.960	0.916	0.933	0.901	0.904
Proposed Method	16	48	0.840	0.885	0.852	0.933	0.827	0.900	0.769	0.815
		96	0.914	0.914	0.880	**0.974**	0.889	**0.954**	0.833	0.868
		512	0.843	0.871	0.899	**0.965**	0.868	**0.960**	0.867	0.891
		1024	0.849	0.915	0.899	0.953	0.881	0.926	0.870	0.816
	48	48	0.900	**0.950**	0.912	**0.983**	0.892	**0.967**	**0.933**	0.813
		96	**0.931**	0.900	0.831	0.960	0.867	0.915	0.785	0.827
		512	**0.969**	**0.975**	**0.942**	**0.987**	**0.930**	**0.980**	0.893	**0.913**
		1024	0.904	0.918	0.886	**0.971**	0.897	**0.970**	0.807	0.863

Advanced Telecommunications Technology Research (SCAT). The computation was carried out using the computer resource offered under the category of Comprehensive Projects between Yamaguchi University and Kyushu University by Research Institute for Information Technology, Kyushu University.

REFERENCES

[1] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[2] C. Saharia, W. Chan, S. Saxena, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in neural information processing systems*, vol. 35, 36479–36494, 2022.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[4] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, "Tree-rings watermarks: Invisible fingerprints for diffusion images," *Advances in Neural Information Processing Systems*, vol. 36, 58047–58063, 2023.

[5] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[8] E. García-Huete, S. Ignacio-Cerrato, D. Pacios, *et al.*, "Evaluating the role of generative ai and color patterns in the dissemination of war imagery and disinformation on social media," *Frontiers in Artificial Intelligence*, vol. 7, 1457247, 2025.

[9] W. Wan, J. Wang, Y. Zhang, J. Li, H. Yu, and J. Sun, "A comprehensive survey on robust image watermarking," *Neurocomputing*, vol. 488, pp. 226–247, 2022.

[10] X. Zhao, K. Zhang, Z. Su, *et al.*, "Invisible image watermarks are provably removable using generative AI," *Advances in neural information processing systems*, vol. 37, pp. 8643–8672, 2024.

[11] K. Chen, "Digital watermarking and steganography," in *Encyclopedia of Multimedia Technology and Networking, Second Edition*, IGI Global, 2009, pp. 402–409.

[12] L. Lei, K. Gai, J. Yu, and L. Zhu, "Diffusetrace: A transparent and flexible watermarking scheme for latent diffusion model," *arXiv preprint arXiv:2405.02696*, 2024.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[15] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.

[16] N. Venkatanath, D. Praneeth, M. C. Bh, S. S. Channappayya, and S. S. Medasani, "Blind image quality evaluation using perception based features," in *2015 twenty first national conference on communications (NCC)*, IEEE, 2015, pp. 1–6.

[17] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[18] Stable Diffusion Web, *Stable diffusion prompts generator*, https : / / stablediffusionweb . com / ja / prompts, Accessed: 2025-06-09, 2025.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *2010 20th international conference on pattern recognition*, IEEE, 2010, pp. 2366–2369.