Robust-Wide: Robust Watermarking against Instruction-driven Image Editing

Runyi Hu[1,2], Jie Zhang[1*], Ting Xu[3], Jiwei Li[4], and Tianwei Zhang[1]

[1] Nanyang Technological University
runyi_hu@163.com
{jie_zhang, tianwei.zhang}@ntu.edu.sg
[2] S-Lab, NTU
[3] National University of Singapore
xuting@nus.edu.sg
[4] Zhejiang University
jiwei_li@zju.edu.cn

Abstract. Instruction-driven image editing allows users to quickly edit an image according to text instructions in a forward pass. Nevertheless, malicious users can easily exploit this technique to create fake images, which could cause a crisis of trust and harm the rights of the original image owners. Watermarking is a common solution to trace such malicious behavior. Unfortunately, instruction-driven image editing can significantly change the watermarked image at the semantic level, making current state-of-the-art watermarking methods ineffective.

To remedy it, we propose `Robust-Wide`, the first robust watermarking methodology against instruction-driven image editing. Specifically, we follow the classic structure of deep robust watermarking, consisting of the encoder, noise layer, and decoder. To achieve robustness against semantic distortions, we introduce a novel Partial Instruction-driven Denoising Sampling Guidance (*PIDSG*) module, which consists of a large variety of instruction injections and substantial modifications of images at different semantic levels. With *PIDSG*, the encoder tends to embed the watermark into more robust and semantic-aware areas, which remains in existence even after severe image editing. Experiments demonstrate that `Robust-Wide` can effectively extract the watermark from the edited image with a low bit error rate of nearly 2.6% for 64-bit watermark messages. Meanwhile, it only induces a neglectable influence on the visual quality and editability of the original images. Moreover, `Robust-Wide` holds general robustness against different sampling configurations and other popular image editing methods such as ControlNet-InstructPix2Pix, MagicBrush, Inpainting, and DDIM Inversion. Codes and models are available at https://github.com/hurunyi/Robust-Wide.

Keywords: Watermarking · Image Editing

[*] The corresponding author

1 Introduction

Recently released Text-to-Image (T2I) diffusion models (e.g., GLIDE [17], DALL.E 2 [20], Imagen [23], Stable Diffusion [21]) have pushed image generation capabilities to a new level. Trained on massive text-image pairs collected from the Internet, these models can generate high-quality photorealistic images based on given text prompts. Instruction-driven image editing, a fantastic technique utilizing the power of T2I diffusion models, can edit the images on demand according to the instructions. Different models have been introduced to achieve this task. For instance, InstructPix2Pix [1] is a popular instruction-driven image editing model, which is fine-tuned from Stable Diffusion [21] on the dataset generated by GPT-3 [3] and Prompt2Prompt [8]. Afterwards, HIVE [32], MagicBrush [29], and MGIE [6] are proposed to improve InstructPix2Pix.

Despite the success of the instruction-driven image editing technique, these models could be misused by malicious users. First, attackers can exploit these models to modify normal images to create fake news, causing a crisis of trust in an individual or even a country. Typical examples include changing someone's face or expression, forging endorsements for commercial gain, or taking off the clothes to produce vulgar images. Second, attackers can migrate the style based on a certain painting or make local modifications while keeping the basic composition of the painting unchanged to create new works, which shall be confirmed as plagiarism, infringing the IP rights of the work's owner. By integrating the personalization technique (e.g., Textual Inversion [7]), they may compose some concepts learned from other images in the edited image, which will undoubtedly lead to wider infringement on the concept's owners.

To identify such misuse and trace malicious users, a common approach is watermarking. We can embed a secret watermark message into the original image, which can be extracted later for ownership verification. The embedded watermark must be robust enough against various distortions. Traditional robust watermarking strategies [19] mainly embed the watermark into a transformed domain to resist spatial distortions. To achieve robustness against more complex digital distortions, some deep watermarking methods are further proposed, e.g., HiDDeN [33] and MBRS [12]. Additionally, researchers also introduced new methods to pursue robustness against physical distortions, including StegaStamp [25] and PIMoG [5]. Unfortunately, all the above solutions mainly target the pixel-level distortions, and fail to resist instruction-driven image editing, which induces significant distortions in the semantic level.

To remedy this issue, we propose `Robust-Wide`, the first robust watermarking method for instruction-driven image editing. We are motivated by the popular encoder-noise layer-decoder framework in most deep watermarking methods [5, 12, 25, 33], which jointly achieve watermark embedding and extraction in an end-to-end way and leveragzhi lie the noise layer to simulate specific distortions to obtain the corresponding robustness. However, the main challenge in our task is how to simulate the distortions caused by instruction-driven image editing. To this end, we design a novel Partial Instruction-driven Denoising Sampling Guidance (*PIDSG*) module in `Robust-Wide`. Briefly, *PIDSG* allows the gradient of the last k sampling steps to flow into the training pipeline, making the non-differentiable sampling process trainable. Additionally, it injects diverse instructions to guide distortions, forcing the encoder and decoder to focus on semantic areas for watermark embedding and extraction.

We perform extensive experiments to demonstrate the robustness of `Robust-Wide` during the instruction-driven image editing process. It achieves a low Bit Error Rate (BER) of nearly 2.6% for 64-bit watermark messages, while preserving the visual quality and editability of original images. Besides the robustness against semantic distortions, `Robust-Wide` acquires inherent robustness against pixel-level distortions such as JPEG and color shifting, which are unseen during training. It also holds general robustness against different sampling configurations and even different popular editing models such as ControlNet-InstructPix2Pix, MagicBrush, Inpainting [10] and DDIM Inversion [16].

In summary, our contributions are as follows:

- We point out the potential threat caused by the misuse of instruction-driven image editing and find current state-of-the-art watermarking methods are ineffective in the emerging case. In other words, we unearth a novel robustness requirement for current image watermarking.
- We propose `Robust-Wide`, the first robust watermarking method for instruction-driven image editing. We introduce a novel Partial Instruction-driven Denoising Sampling Guidance (*PIDSG*) module, which forces the watermark embedded in the semantic-level rather than pixel-level.
- Experimental results demonstrate that `Robust-Wide` can resist instruction-driven image editing, conventional pixel-level distortions, and different sampling configurations. More importantly, the proposed method exhibits robustness against a variety of popular editing models.

2 Background

2.1 Diffusion Model

Inspired by the non-equilibrium statistical physics, Diffusion Model (DM) [24] destroys the structure in a data distribution through an iterative forward diffusion process, and learns a reverse diffusion process to restore data's structure. Denoising Diffusion Probabilistic Model (DDPM) [9] further improves the performance of DM by training on a weighted variational bound with the folllowing objective:

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right], \tag{1}$$

where x is the input image, ϵ is the randomly sampled Gaussian noise, $t \in \{1, ..., T\}$ is the uniformly sampled timestep, x_t is the noisy version of x, and ϵ_θ is the diffusion model trained to predict a denoised variant of x_t.

Many T2I diffusion models, e.g., GLIDE [17], DALL·E 2 [20] and Imagen [23] are based on DDPM. They operate directly in the pixel space, which can consume a large amount of computational resources for both training and evaluation. To overcome these shortcomings, Latent Diffusion Model (LDM) [21] is proposed to perform the noise and denoise process in the latent space of the pre-trained VAE:

$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right], \tag{2}$$

where $\mathcal{E}$ is the encoder of VAE and z_t is the noisy latent variable. Stable Diffusion is a popular T2I model based on LDM with great impact and outstanding performance.

2.2 Instruction-driven Image Editing

There are mainly five popular instruction-driven image editing models, which are all based on DMs. (1) InstructPix2Pix [1] performs editing in a forward pass quickly and does not require any user-drawn mask, additional images, per-example fine-tuning, or inversion. It is trained in an end-to-end manner, with the dataset generated by GPT-3 [3] and Prompt2Prompt [8]. Each item in the dataset contains a source image, an editing instruction, and a subsequent edited image. During training, the image and instruction are regarded as conditions c_I and c_T, respectively, while the edited image is the ground-truth output. Therefore, the training object is as follows:

$$L = \mathbb{E}_{\mathcal{E}(x),\mathcal{E}(c_I),c_T,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \mathcal{E}(c_I), c_T)\|_2^2\right]. \tag{3}$$

(2) ControlNet [31] is a dedicated framework that amplifies the capability and controllability of pre-trained T2I diffusion models by integrating spatial conditioning controls. Therefore, we can regard original images as spatial conditioning controls and train ControlNet on the dataset of InstructPix2Pix to realize instruction-driven image editing, which is called ControlNet-InstructPix2Pix [31]. (3) Afterwards, HIVE [32] improves InstructPix2Pix by harnessing human feedback to tackle the misalignment between editing instructions and resulting edited images. (4) MagicBrush [29] introduces the first large-scale and manually annotated dataset for instruction-guided real image editing and fine-tunes InstructPix2Pix on the dataset for better performance. (5) MGIE [6] uses multimodal large language models to derive expressive instructions and provide explicit guidance to further improve the editing performance while maintaining the efficiency. Besides, there are also some popular text-driven image editing methods such as Inpainting [10] and DDIM Inversion [16]. Inpainting edits an image by masking some regions of it and then regenerates the image according to the given text. DDIM Inversion [16] performs the reversed DDIM sampling process conditioned on the original text caption to convert the image to its partially noised version. Afterwards, the edited image is acquired by denoising the noised image given the edited text caption.

In this paper, we mainly target InstructPix2Pix, and assess the generalization of our method to ControlNet-InstructPix2Pix, MagicBrush, Inpainting and DDIM Inversion based on Stable Diffusion, since these models are open-sourced and publicly available.

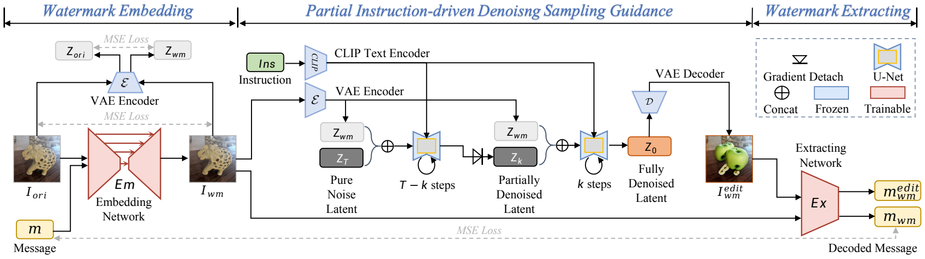

Fig. 1: The overall training pipeline of `Robust-Wide`.

2.3 Robust Watermarking

Robust watermarking is widely used for IP protection and forensics. Traditional methods (*e.g.*, DWT-DCT [19] and DWT-DCT-SVD [19]) embed a pre-defined watermark message into transformed domains to achieve simple robustness against different operations, e.g., affine transformation, scaling, etc. HiDDeN [33] is the first to leverage deep neural networks for watermark embedding and extraction. Importantly, it simulates JPEG compression into a differential module, inserted into the famous encoder-noise layer-decoder framework for robustness enhancement. RivaGAN [30] applies a customized attention-based mechanism to embed diverse data and includes a separate adversarial network for optimizing robustness. Afterwards, MBRS [12] utilizes a mixture of real JPEG, simulated JPEG, and noise-free images to get superior robustness performance against JPEG compression. Similarly, CIN [14] combines invertible and non-invertible mechanisms for robustness to various digital distortions. In addition to digital robustness, numerous works (LFM [27], StegaStamp [25], RIHOOP [11], PIMoG [5]) try to acquire robustness against physical operations such as screen-shooting, print-shooting, etc. Recently, SepMark [28] utilizes a unified framework for source tracing and Deepfake detection which also achieves robustness against face manipulations, a special form of image editing. However, none of them can resist instruction-driven image editing (see Table 1.)

In this paper, we compare our method with current state-of-the-art watermarking methods such as MBRS [12], CIN [14], PIMoG [5], and SepMark [28]. We also adopt DWT-DCT [19], DWT-DCT-SVD [19], and RivaGAN [30] as the baseline methods, which are suggested by Stable Diffusion's official webpage [26].

3 Methodology

3.1 Overview

We introduce `Robust-Wide`, a novel methodology to embed robust watermarks into images, which can resist instruction-driven editing. Its overall training pipeline is shown in Figure 1. An embedding network E_m and extracting network E_x are jointly trained to achieve watermark embedding and extraction, respectively. Furthermore, a novel Partial Instruction-driven Denoising Sampling Guidance ($PIDSG$) module is integrated into the pipeline to enhance the watermark robustness against instruction-driven image editing. With the trained networks, we can use E_m to embed a secret watermark message into a protected image, and release it to public. If a malicious user performs instruction-driven image editing over this watermarked image without authorization, we are able to detect this misuse by using E_x to extract the watermark message from the edited image. Below we describe each step in detail.

3.2 Watermark Embedding

An embedding network E_m is introduced to generate a watermarked image I_{wm} from the original image I_{ori}, where the watermark is a random L-bit message $m \in \{0,1\}^L$. Specifically, we adopt U-Net [22] as the structure of E_m. To match m with the dimension of I_{ori}, E_m converts the flattened version of m in the shape of $1 \times \sqrt{L} \times \sqrt{L}$ to the shape of $C \times H \times W$ using some transposed convolutional layers, where C is the predefined feature channel, H and W are the height and weight of I_{ori}. Then, we concatenate the reshaped message with I_{ori}. To preserve the editing capability of the image, I_{wm} should be visually consistent with I_{ori}. We first adopt the L_2 distance between I_{ori} and I_{wm} in the pixel level, *i.e.*,

$$L_{em_1} = L_2(I_{ori}, I_{wm}) = L_2(I_{ori}, E_m(I_{ori}, m)). \tag{4}$$

Furthermore, we add another constraint between I_{ori} and I_{wm} in the feature space, represented by the encoder $\mathcal{E}$ of VAE in InstructPix2Pix, *i.e.*,

$$L_{em_2} = L_2(Z_{ori}, Z_{wm}) = L_2(\mathcal{E}(I_{ori}), \mathcal{E}(I_{wm})). \tag{5}$$

3.3 Partial Instruction-driven Denoising Sampling Guidance

Existing watermarking solutions mainly consider the robustness against pixel-level distortions. In contrast, instruction-driven image editing changes an image significantly at the semantic level, making these approaches ineffective. We note that instruction-driven image editing involves the injection of a large number of different semantic instructions and varying degrees of image modifications at different semantic levels, which can be utilized to guide the robust and semantic-aware watermark embedding and extraction process. Thus, our intuitive idea is to incorporate the editing process into the end-to-end training framework. However, one challenge is that during the denoising sampling process (*e.g.*, in InstructPix2Pix), gradients are not allowed to flow directly. Introducing this process into training would result in the inability of gradients to propagate from the watermark decoder through the sampling process back to the watermark encoder. In other words, this would lead to a discontinuity in the computational graph, rendering the entire process non-differentiable. While a straightforward approach might be to open gradients for the numerous denoising sampling steps, this would introduce a significant memory overhead. To address this, we design the Partial Instruction-driven Denoising Sampling Guidance (*PIDSG*) module, which selectively allows gradients to flow only in the last k sampling steps. This design not only makes the entire method feasible but also enables the process to be differentiable and amenable to end-to-end optimization.

As shown in the middle part of Figure 1, InstructPix2Pix consists of VAE [4], U-Net [22], and CLIP text encoder [18]. We freeze all the parameters of these models. During training, the encoder $\mathcal{E}$ of VAE converts I_{wm} to its latent $Z_{wm} = \mathcal{E}(I_{wm})$. Then, Z_{wm} is concatenated with the pure noise latent Z_T and sent to U-Net to perform denoising sampling iterations. Assuming the sampling process totally has T steps, we truncate the gradient flow in the first $T-k$ steps to obtain the partially denoised latent Z_k. After that, we concatenate Z_k with Z_{wm} and perform the last k sampling steps (dubbed gradient backward steps) to enable the gradient flow. The CLIP text encoder processes the instruction Ins and outputs the textual embedding to guide the whole sampling process. Finally, after T sampling steps, the fully denoised latent Z_0 is produced and converted to the edited image I_{wm}^{edit} by the decoder $\mathcal{D}$ of VAE.

3.4 Watermark Extracting

For the extracting network E_x, we leverage some residual blocks as its architecture. With the edited image I_{wm}^{edit}, E_x aims to extract the message m_{wm}^{edit} that is consistent to the original embedded message m, *i.e.*,

$$L_{ex_1} = MSE(m, m_{wm}^{edit}) = MSE(m, E_x(I_{wm}^{edit})). \tag{6}$$

For effective forensic, we also require E_x to be capable of extracting the embedded watermark message m_{wm} from the watermarked image I_{wm} before editing:

$$L_{ex_2} = MSE(m, m_{wm}) = MSE(m, E_x(I_{wm})). \tag{7}$$

Interestingly, we observe that the training will not converge without L_{ex_2}. We explain that the extracting network E_x cannot find the watermark area if only fed with edited images that are different from the original images at the semantic level. More results can be found in Sec. 4.5.

3.5 Joint Training

We jointly train E_m and E_x with the above-mentioned components: watermark embedding, *PIDSG*, and watermark extraction. The total loss is formulated as follows:

$$L_{total} = L_{em_1} + \lambda_1 L_{em_2} + \lambda_2 L_{ex_1} + \lambda_3 L_{ex_2}, \tag{8}$$

where $\lambda_1 = 0.001$, $\lambda_2 = 0.1$, and $\lambda_3 = 1$ by default are the hyperparameters to balance each loss item. More analysis can be found in Sec. 4.5.

4 Experiments

Datasets. To train the embedding and extracting networks, we adopt 20k image-instruction pairs from the dataset used in InstructPix2Pix. We also select 1.2k additional samples that do not overlap with the above training data for evaluation by default. Besides, we collect some real-world images from the Internet, which cover 6 types (*i.e.*, person, animal, object, architecture, painting, and scenery) and each type has 5 images which can be found in the supplementary material. Then, we use InstructPix2Pix [2] by default to edit these images based on 6 instructions and generate 8 images per instruction to finally obtain 1.44k edited images for testing.

Implementation Details. We train all our models on a single A6000 GPU, with a learning rate of 1e-3, batch size of 2, and total steps of 20,000. We use the AdamW optimizer with a cosine scheduler of 400 warm-up steps. Images used for training and evaluation are all 512×512 by default. For the configurations of *PIDSG*, we adopt the Euler sampler, with 20 inference steps, the text guidance scale s_T=10.0, and image guidance scale s_I=1.5.

We choose seven baselines for comparisons, *i.e.*, DWT-DCT [19], DWT-DCT-SVD [19], RivaGAN [30], MBRS [12], CIN [14], PIMoG [5] and SepMark [28]. We directly use their official code for implementation. Noteably, the released model of MBRS and SepMark only support 256×256 images with 256 bits and 128 bits separately and we used the Tracer of SepMark for watermark extraction after image editing. Since the released model of CIN only supports 128×128 images and it is robust against the resize operation, so we resized the watermarked image to 256×256 for image editing and then resized it back to 128×128 for watermark extraction. PIMoG does not provide model weights, we trained the model ourselves using the official code.

Metrics. To evaluate the effectiveness of our method, we measure the Bit Error Rate (BER) between the extracted watermark X and ground-truth watermark Y, *i.e.*, $BER(X,Y) = \frac{\sum_{i=1}^{L}(X_i \neq Y_i)}{L}$, where $X_i, Y_i \in \{0,1\}$ and L is the watermark length. To assess the fidelity, we adopt PSNR and SSIM to measure the visual quality of watermarked images. To verify how the watermarked image can preserve its original editability, we adopt the CLIP image similarity (CLIP-I) and CLIP Text-Image Direction Similarity (CLIP-T), which are also used in InstructPix2Pix [1].

4.1 Effectiveness Evaluation

Table 3: The integrity of `Robust-Wide` and the influence on the image editability.

Metrics	w/o Editing (**BER** (%) ↓)	w/ Editing (**BER** (%) ↓)	**CLIP-I** ↑	**CLIP-T** ↑
Original Images	49.8403	48.8550	0.8402	0.2183
Watermarked Images	0.0000	2.6579	0.8430	0.2148

Table 1 compares the effectiveness of `Robust-Wide` with the baseline methods with different image sizes and watermark lengths. For `Robust-Wide`, we consider the implementation without and with *PIDSG*. From this table, it is obvious that none

Table 1: Quantitative results compared with other methods.

Method	Image Size	Watermark Length (bits)	BER (%) ↓		PSNR↑	SSIM↑
			w/o Editing	w/ Editing		
DWT-DCT [19]	512x512	32	11.9351	49.2286	38.7123	0.9660
DWT-DCT-SVD [19]	512x512	32	0.0314	47.5680	38.6488	0.9726
RivaGAN [30]	512x512	32	0.6276	40.5256	40.6132	0.9718
MBRS [12]	256x256	256	0.0000	46.7661	43.9780	0.9870
CIN [14]	128x128	30	0.0000	44.9888	40.3678	0.9763
PIMoG [5]	256x256	64	0.0000	49.9635	35.3183	0.9212
SepMark [28]	256x256	128	0.0084	28.1460	36.4341	0.9194
`Robust-Wide`	512x512	64	0.0000	2.6579	41.9142	0.9910
`Robust-Wide`	512x512	256	0.0000	4.1867	39.1842	0.9844

Table 2: The importance of *PIDSG*. † denotes the results on real-world images.

Method	Image Size	Watermark Length (bits)	BER (%) ↓		PSNR↑	SSIM↑
			w/o Editing	w/ Editing		
`Robust-Wide` (w/o *PIDSG*)	512x512	64	0.0000	50.1558	55.3710	0.9982
`Robust-Wide` (w/ *PIDSG*)	512x512	64	0.0000	2.6579	41.9142	0.9910
`Robust-Wide` (w/o *PIDSG*) †	512x512	64	0.0000	51.0801	55.3715	0.9983
`Robust-Wide` (w/ *PIDSG*) †	512x512	64	0.0000	2.6062	41.4038	0.9922

of the baseline methods can resist instruction-driven image editing, with the BER of around 50%. Comparably, `Robust-Wide` is effective with a low BER of 2.6579%. Table 2 also shows the effectiveness of `Robust-Wide` on 1.44k real-world samples. Importantly, the removal of *PIDSG* leads to a complete failure, revealing the importance of *PIDSG*. Moreover, Table 3 shows that `Robust-Wide` will not extract watermarks from original images with or without editing, guaranteeing forensic integrity.

4.2 Fidelity Evaluation

Table 1 also shows the PSNR and SSIM of different methods. We observe that `Robust-Wide` can achieve the comparable visual quality with other baseline methods. Table 3 compares the values of CLIP-I and CLIP-T for original and watermarked images. The slight difference indicates that `Robust-Wide` induces little influence on the editability. Figure 2 shows some visual results using our `Robust-Wide`, which further confirms the fidelity (more results are provided in the supplementary material). Specifically, the normalized residual images are computed as $N(|I_{wm} - I_{ori}|)$, where $N(x) = (x - min(x))/(max(x) - min(x))$. From these images, it is evident that the watermark is predominantly embedded along the contours of the main subjects (such as people or objects) and in the background (secondary elements like buildings). We posit that these areas may be robust regions related to conceptual content and therefore `Robust-Wide` tends to embed watermarks into robust concept-aware areas.

4.3 Robustness Evaluation

Pixel-level Distortions. We apply different pixel-level distortions in three

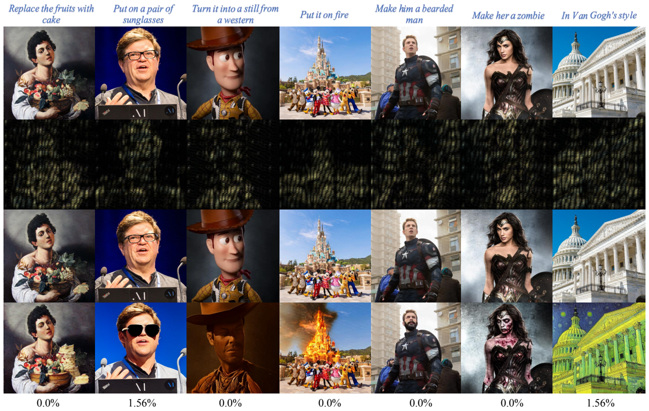

Fig. 2: Visual results for `Robust-Wide`. From top to bottom: instructions, original images, normalized residual images, watermarked images, edited images, and the corresponding BERs.

Table 4: Robustness of `Robust-Wide` against pixel-level distortions.

BER(%)	None	JPEG	Median Blur	Gaussian Blur	Gaussian Noise	Sharpness	Brightness	Contrast	Saturation	Hue	Noise+Denoise
I	2.6579	2.7256	2.4926	2.7074	3.0672	2.6150	12.2907	5.1336	3.0455	2.7591	8.6401
II	0.0000	0.0013	0.0000	0.0013	0.0716	0.0013	0.4733	0.9210	0.0000	0.0000	3.9071
III	2.6579	2.7934	2.9574	2.8489	6.0546	2.7773	9.4796	4.3346	3.0373	3.2829	9.3654

ways: (I) pre-processing watermarked images before editing; (II) post-processing watermarked images; (III) post-processing edited images based on watermarked images. Table 4 shows the extraction error of `Robust-Wide` against different pixel-level distortion types in three scenarios. It is obvious that `Robust-Wide` demonstrates strong robustness against these distortions even if we do not involve them during training.

Additionally, we also test `Robust-Wide`'s robustness against DiffWA [13], a new watermark removal attack. This attack utilizes an image-to-image conditional diffusion model to add noise on the watermarked image and then restores the image while removing the embedded watermark. Due to the unavailability of DiffWA's source code, we utilize the open-source SDEdit [15] from Diffusers to carry out the process of adding noise and subsequently restoring the watermarked images. We configure the sampling parameters as follows: strength=0.2, prompt=Null, with all other sampling parameters as their default values. As shown in the "Noise+Denoise" row of Table 4, `Robust-Wide` is effective in all three situations with BER$< 10\%$.

Different Sampling Configurations. We assess the robustness of `Robust-Wide` against different diffusion sampling configurations. Figure 3 shows the corresponding results. We have the following observations. (1) `Robust-Wide` is generally effective for different samplers. (2) As the total number of inference steps increases, the edited images become more fine-grained while the BER slightly increases (within a range of 1%). We explain that more details lead to larger differences between the edited and original images. (3) A larger text guidance

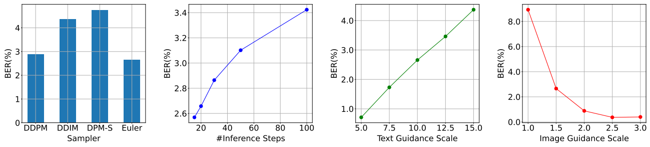

Fig. 3: Robustness of `Robust-Wide` against different diffusion sampling configurations.

scale (s_T) and smaller image guidance scale (s_I) indicate more severe image editing. `Robust-Wide` achieves BER of below 5% in all cases except when s_I is 1, validating its general robustness in different settings.

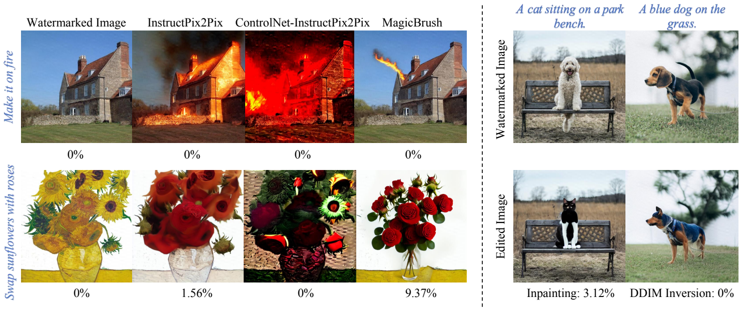

Fig. 4: General robustness against other editing methods such as InstructPix2Pix, ControlNet-InstructPix2Pix, MagicBrush, Inpainting, and DDIM Inversion.

Other Popular Image Editing Methods. In addition to InstructPix2Pix, we further evaluate our robustness against its extension ControlNet-InstructPix2Pix and MagicBrush on the 1.2k-samples dataset. Experiments show that `Robust-Wide` can achieve an average BER of 0.96% on ControlNet-InstructPix2Pix and 9.34% on MagicBrush. Figure 4 shows some visual examples (more examples are provided in the supplementary material). For the instruction "swap sunflowers with roses", MagicBrush makes the edited image more different from the original image, leading to a relatively higher but still acceptable BER.

We also tested the robustness of Robust-Wide against Inpainting [10] and DDIM Inversion [16] based on Stable Diffusion Models. We found that, even though Robust-Wide has never seen these image editing methods during training, it still demonstrates effective resistance as shown in Figure 4.

Continual Editing. A user may utilize InstructPix2Pix to perform multiple edits on a single image. An image edited by one user could also be spread to another for further editing, and this process can repeat several rounds. Hence, we need to ensure our watermarks can resist continual editing. As shown in Figure 5, we observe that the watermark embedded into the original image can be accurately extracted even after 3 editing rounds, demonstrating the watermark's strong robustness against continual editing.

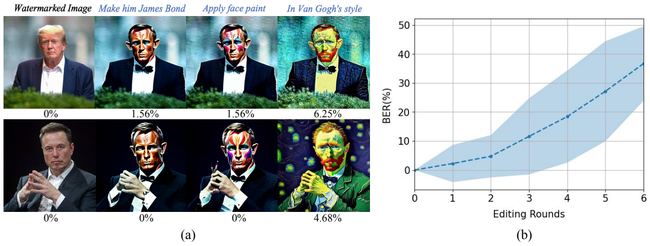

Fig. 5: The influence of continual editing. (a) Some visual examples under continual editing (from left to right). (b) BER increases with more editing rounds. This experiment is conducted on real-world images as mentioned above.

4.4 More Analysis

Embedding mode of `Robust-Wide`. Figure 6 displays the normalized residual image between the watermarked and original images, using different watermarking methods. We can see that the watermarks embedded using DWT-DCT and DWT-DCT-SVD are quite faint and imperceptible, indicating their limited robustness. MBRS tends to concentrate primarily along the object's edge contours, making it vulnerable to changes in the image background. In contrast, RivaGAN and our `Robust-Wide` embed watermarks in both the edges and backgrounds, resulting in enhanced robustness. Furthermore, we use Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT) to visualize the residual images in the frequency domain. From the residual images, we can observe that DWT-DCT and DWT-DCT-SVD introduce input-agnostic modification while other methods introduce input-aware modification. From the last two rows of Figure 6, we find that DWT-DCT, DWT-DCT-SVD, and RivaGAN tend to mainly modify the low-frequency area while MBRS, CIN, PIMoG, and SpeMark prefer to embed the watermark into both low-frequency and high-frequency areas. Differently, `Robust-Wide` mainly focuses on embedding watermarks in infra-low frequency areas or mid-low frequency areas, potentially making it more robust against semantically and conceptually related image modifications.

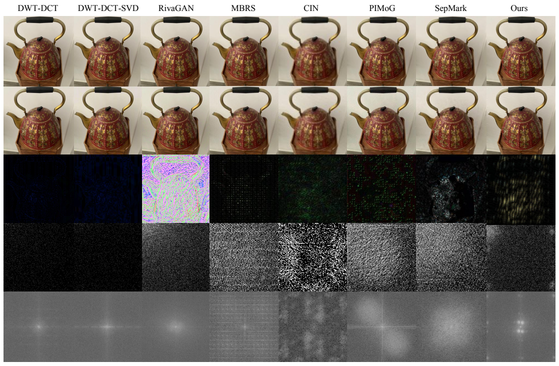

Fig. 6: The embedding mode of different methods. From top to bottom: original images, watermarked images, normalized residual images, DCT and DFT of residual images, respectively

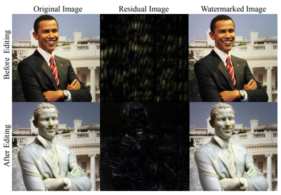

Fig. 7: Example of our extracting mode.

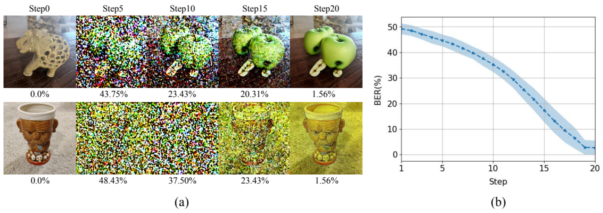

Fig. 8: The influence of the number of sampling steps. (a) Some visual examples at different steps and their corresponding BER. (b) BER decreases with more steps.

Extracting mode of `Robust-Wide`. We use the original image I_{ori} and watermarked image I_{wm} to generate the corresponding edited images I_{ori}^{edit} and I_{wm}^{edit}. Here, we control all potential random factors (*e.g.*, generation random seed) to guarantee that the difference between I_{ori}^{edit} and I_{wm}^{edit} is solely caused by instruction-driven image editing. As shown in Figure 7, the watermark embedded into the original image also exists after editing, which preserves as an outline of the character. We hypothesize that the extractor mainly focuses on such robust concept-aware areas. More examples are provided in the supplementary material.

Relationship between the number of sampling steps and extraction ability. We test the BER of generated images at different sampling steps. Figure 8 (a) indicates that the model focues on generating contours and layout at the first few sampling steps, and then optimizing image details at later steps. As shown in Figure 8 (b), the BER decreases with more sampling steps.

Relationship between editing strength and extraction ability. We explore the impact of the image editing strength (quantified by PSNR and SSIM computed between I_{wm} and I_{wm}^{edit}) on the extraction BER. As shown in Figure 9, points with higher BER are mainly at the area where PSNR and SSIM are low, *i.e.*, the editing strength is large. This is intuitive as the greater the change is, the more difficult it will be for the watermark extraction. When the editing strength on the original image is significant, the resulting edited image can be considered as a form of re-creation. In such cases, the risk of copyright infringement is relatively low, and the ineffective extraction may be deemed acceptable.

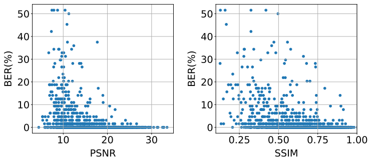

Fig. 9: The relationship between image editing strength and extraction ability.

4.5 Ablation Study

Importance of L_{ex_2}. With L_{ex_2}, the embedding network and extracting network tend to find desirable watermarking areas at first and then search more robust areas to resist instruction-driven image editing. Without L_{ex_2}, it is challenging to achieve watermark embedding and extraction only with edited images.

Table 5 compares the performance under these two settings. The embedded watermark cannot be effectively extracted without L_{ex_2}, both before and after editing, with the BER of around 50%. Hence, L_{ex_2} is essential for the overall effectiveness of `Robust-Wide`.

Impact of hyper-parameters λ_1 and λ_2. Table 6 shows the watermark performance with different λ_1 and λ_1 values. We observe that a larger λ_1 can help improve the visual quality of watermarked images but causes a decrease in the watermark extraction rate. On the other hand, a higher λ_2 leads to lower BER, which showcases the trade-off between extraction ability and visual quality.

Table 5: The importance of L_{ex_2}. The gray cell denotes the default setting.

Metrics		w/o L_{ex_2}	w/ L_{ex_2}
BER(%)↓	w/o editing	50.1098	0.0000
	w/ editing	50.1219	2.6579
PSNR↑		68.7220	41.9142
SSIM↑		0.9999	0.9910

Table 6: The impact of different λ_1 and λ_2.

Metrics	λ_1				λ_2		
	0	0.001	0.01	0.1	0.01	0.1	1
BER(%)↓	1.7391	2.6579	6.0690	50.0364	11.1796	2.6579	0.9674
PSNR↑	39.6859	41.9142	41.7604	56.9096	45.8701	41.9142	35.2577
SSIM↑	0.9750	0.9910	0.9938	0.9986	0.9929	0.9910	0.9616

Table 7: The influence of k and the bits length.

Metrics	k			**Bits Length**			
	1	2	3	16	64	256	1024
BER(%)↓	4.0520	2.9166	2.6579	2.2812	2.6579	4.1867	6.3036
PSNR↑	44.3330	42.1386	41.9142	40.8327	41.9142	39.1842	36.5956
SSIM↑	0.9938	0.9919	0.9910	0.9853	0.9910	0.9844	0.9732

Influence of the number of gradient backward steps k. Table 7 shows the watermark performance with different numbers of gradient backward steps k. With more steps, the watermark message is easier to extract while the visual quality of the watermarked image slightly reduces. Due to the GPU memory constraints, the maximum number of gradient backward steps we could set is 3, which is sufficient to obtain acceptable performance.

Influence of different watermark bits length. Table 7 reports the evaluation results with different lengths of watermark messages. A longer watermark message results in higher BER and lower visual quality. In practice, the user can customarily select the watermark lengths to balance such trade-off. In Table 7, we can observe that the visual quality is lower when the watermark length is 16 compared to when the watermark length is 64. Indeed, the integration of watermark bit with the image involves shape transformations through convolutional or deconvolutional layers. With 16 bits, our available GPU memory posed restrictions on employing convolutional or deconvolutional layers for shape transformations. Consequently, we opted for a non-parametric interpolation method to handle the integration, causing degradation in visual quality.

5 Conclusion

In this paper, we propose `Robust-Wide`, the first robust watermarking method against instruction-driven image editing. Our core idea is to integrate a novel module called *PIDSG* into the encoder-noise layer-decoder training framework, which forces watermarks embedded in the semantic level. Experiments demonstrate that `Robust-Wide` is robust against different image editing methods while maintaining high visual quality and editability. Furthermore, our in-depth analysis on the embedding and extracting modes of `Robust-Wide` is expected to shed light on the design of watermarking against other semantic distortions.

Acknowledgements

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s), and Singapore Ministry of Education (MOE) AcRF Tier 2 MOE-T2EP20121-0006.

References

1. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18392–18402 (June 2023) 2, 4, 8
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions (2023) 8
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf 2, 4
4. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021) 7
5. Fang, H., et al.: Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In: ACM MM. pp. 2267–2275 (2022) 2, 5, 8, 9
6. Fu, T.J., Hu, W., Du, X., Wang, W.Y., Yang, Y., Gan, Z.: Guiding instruction-based image editing via multimodal large language models (2023) 2, 4
7. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. In: The Eleventh International Conference on Learning Representations (2022) 2
8. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-or, D.: Prompt-to-prompt image editing with cross-attention control. In: The Eleventh International Conference on Learning Representations (2022) 2, 4
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf 3
10. stable-diffusion inpainting: https://huggingface.co/runwayml/stable-diffusion-inpainting 3, 4, 11
11. Jia, J., Gao, Z., Chen, K., Hu, M., Min, X., Zhai, G., Yang, X.: Rihoop: Robust invisible hyperlinks in offline and online photographs. IEEE Transactions on Cybernetics **52**(7), 7094–7106 (2020) 5

12. Jia, Z., Fang, H., Zhang, W.: Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In: Proceedings of the 29th ACM international conference on multimedia. pp. 41–49 (2021) 2, 5, 8, 9
13. Li, X.: Diffwa: Diffusion models for watermark attack (2023) 10
14. Ma, R., Guo, M., Hou, Y., Yang, F., Li, Y., Jia, H., Xie, X.: Towards blind watermarking: Combining invertible and non-invertible mechanisms. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 1532–1542 (2022) 5, 8, 9
15. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=aBsCjcPu_tE 10
16. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023) 3, 4, 11
17. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., Chen, M.: GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 16784–16804. PMLR (17–23 Jul 2022), https://proceedings.mlr.press/v162/nichol22a.html 2, 3
18. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 7
19. Rahman, M.M.: A dwt, dct and svd based watermarking technique to protect the image piracy. International Journal of Managing Public Sector Information and Communication Technologies **4**(2), 21 (2013) 2, 5, 8, 9
20. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents (2022) 2, 3
21. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 2, 4
22. Ronneberger, O., Fischer, P., Brox, T.: 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. pp. 234–241. Springer (2015) 6, 7
23. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 36479–36494. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/ec795aeadae0b7d230fa35cbaf04c041-Paper-Conference.pdf 2, 3
24. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Bach, F., Blei, D. (eds.)

Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 2256–2265. PMLR, Lille, France (07–09 Jul 2015), https://proceedings.mlr.press/v37/sohl-dickstein15.html 3

25. Tancik, M., Mildenhall, B., Ng, R.: Stegastamp: Invisible hyperlinks in physical photographs. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2117–2126 (2020) 2, 5
26. invisible watermark, S.: https://github.com/ShieldMnt/invisible-watermark 5
27. Wengrowski, E., Dana, K.: Light field messaging with deep photographic steganography. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1515–1524 (2019) 5
28. Wu, X., Liao, X., Ou, B.: Sepmark: Deep separable watermarking for unified source tracing and deepfake detection. In: Proceedings of the 31st ACM International Conference on Multimedia. p. 1190–1201. MM '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3581783.3612471, https://doi.org/10.1145/3581783.3612471 5, 8, 9
29. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing. arXiv preprint arXiv:2306.10012 (2023) 2, 4
30. Zhang, K.A., Xu, L., Cuesta-Infante, A., Veeramachaneni, K.: Robust invisible video watermarking with attention. arXiv preprint arXiv:1909.01285 (2019) 5, 8, 9
31. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023) 4
32. Zhang, S., Yang, X., Feng, Y., Qin, C., Chen, C.C., Yu, N., Chen, Z., Wang, H., Savarese, S., Ermon, S., Xiong, C., Xu, R.: Hive: Harnessing human feedback for instructional visual editing (2023) 2, 4
33. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018) 2, 5