END^2: Robust Dual-Decoder Watermarking Framework Against Non-Differentiable Distortions

Nan Sun[1], Han Fang[2], Yuxing Lu[3], Chengxin Zhao[1], Hefei Ling[1]*

[1] Huazhong University of Science and Technology
[2] National University of Singapore
[3] Peking University
{sunnan, chengxinzhao, lhfei}@hust.edu.cn, fanghan@nus.edu.sg, yxlu0613@gmail.com

Abstract

DNN-based watermarking methods have rapidly advanced, with the "Encoder-Noise Layer-Decoder" (END) framework being the most widely used. To ensure end-to-end training, the noise layer in the framework must be differentiable. However, real-world distortions are often non-differentiable, leading to challenges in end-to-end training. Existing solutions only treat the distortion perturbation as additive noise, which does not fully integrate the effect of distortion in training. To better incorporate non-differentiable distortions into training, we propose a novel dual-decoder architecture (**END^2**). Unlike conventional END architecture, our method employs two structurally identical decoders: the Teacher Decoder, processing pure watermarked images, and the Student Decoder, handling distortion-perturbed images. The gradient is backpropagated only through the Teacher Decoder branch to optimize the encoder thus bypassing the problem of non-differentiability. To ensure resistance to arbitrary distortions, we enforce alignment of the two decoders' feature representations by maximizing the cosine similarity between their intermediate vectors on a hypersphere. Extensive experiments demonstrate that our scheme outperforms state-of-the-art algorithms under various non-differentiable distortions. Moreover, even without the differentiability constraint, our method surpasses baselines with a differentiable noise layer. Our approach is effective and easily implementable across all END architectures, enhancing practicality and generalizability.

Introduction

With the rapid growth of the Internet, accessing vast digital media resources has become easy, but this also increases the risk of unauthorized use, making protection and ownership verification crucial. In order to verify ownership, digital watermarking technology was first introduced (van Schyndel, Tirkel, and Osborne 1994). Such techniques have been widely studied in image (Hsu and Wu 1999; Hernandez, Amado, and Perez-Gonzalez 2000; Bi et al. 2007), video (Cox et al. 1997; Langelaar, Setyawan, and Lagendijk 2000; Wang and Pearmain 2006) and audio (Swanson et al. 1998; Haitsma et al. 2000; Bassia, Pitas, and Nikolaidis 2001). In recent years, DNN-based image watermarking (Kandi, Mishra, and Gorthi 2017; Ahmadi et al. 2020; Zhu et al. 2018; Tancik, Mildenhall, and Ng 2020) has developed rapidly, with a series of models based on the "Encoder-Noise Layer-Decoder" (END) framework being the most widely used. The END framework employs an autoencoder-like architecture which contains an encoder, a noise layer, and a decoder, as shown in Figure 1 (a). The encoder embeds the watermark information into the cover image, while the decoder extracts the hidden information from the image. The noise layer enhances the model's robustness by applying various distortions. To enable end-to-end joint training, the noise layer must be differentiable. However, real-world distortions are typically non-differentiable (such as JPEG compression and photo filters in third party apps). This limitation significantly reduces the realistic effectiveness of existing models. Therefore, addressing the training for non-differentiable distortions is a crucial step toward developing a practical watermarking framework.

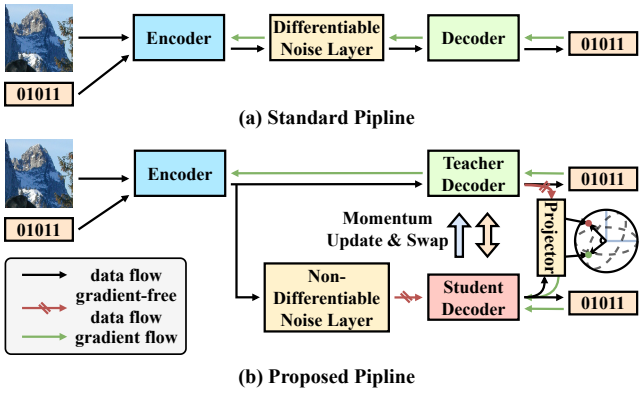

Figure 1: The structures for different methods. (a) The standard END structure, which requires a differentiable noise layer to maintain the joint optimization of the model. (b) Our proposed END^2 structure. Green arrows represent the direction of propagation of the gradient in backpropagation and the red arrow represents that the process is gradient-free.

Current methods addressing non-differentiable distortions typically use mathematical modeling to create differentiable approximations (Zhu et al. 2018; Tancik, Mildenhall, and Ng 2020; Li, Liao, and Wu 2024), which fail to accurately replicate real-world distortions and are ineffective against black-

*Corresponding Author

box distortions with unknown implementations. TDSL (Liu et al. 2019) uses a two-stage training strategy, training the model without noise initially and fine-tuning only the decoder with real noise later. However, the lack of joint training between the encoder and decoder reduces robustness to actual noise effects. In addition, Forward ASL (Zhang et al. 2021) treats all noise as additive and considers only lossless watermarked images during backpropagation. But treating all distortion as additive noise is too idealized to realistically reflect the effect of distortion on the model.

The above methods fail to effectively incorporate the impact of real-world noise during the training process, resulting in a gap between the model's learned behavior and the actual distortions encountered. To address this problem, we propose a novel end-to-end dual-decoder architecture (**END^2**). The Figure 1 (b) illustrates our proposed architecture, which includes an encoder, a noise layer that does not require differentiability, and two decoders. As shown in the figure, in order to bypass the non-differentiable restriction, We utilize the Teacher Decoder branch to propagate the gradient of the clean encoded image directly to the encoder, facilitating the coordinated optimization of both the encoder and decoder. To effectively incorporate non-differentiable distortions and be robust against them, we introduce a Student Decoder, which shares the same structure as the Teacher Decoder but is specifically designed to process distorted images. By maximizing the cosine similarity between the feature vectors of the two decoders in the latent space, the model aligns the representations of distorted and clean images, thereby improving its robustness to non-differentiable distortions.

In addition, we propose using swapping learning and momentum updating strategies to allow the two decoders to supervise each other, which ensures consistent feature representation, enhancing robustness to distortions.

The main contributions of this paper are as follows:

- We propose END^2, a novel dual-decoder framework that effectively addresses the challenges posed by arbitrary non-differentiable distortions.
- Our approach is not only perfectly compatible with differentiable noise layers but also exhibits strong architectural generalization, making it applicable to various END architectures.
- Extensive experiments demonstrate that our method outperforms SOTA methods, exhibiting strong robustness to conventional non-differentiable noise as well as black-box distortions.

Related Work

Deep Learning for Image Watermarking. Recent advances in deep learning have significantly impacted the field of digital image watermarking, harnessing the powerful feature extraction capabilities of neural networks to improve robustness and visual quality. HiDDeN (Zhu et al. 2018) pioneeres an end-to-end DNN-based watermarking framework with an autoencoder-like architecture, which has a profound impact on subsequent models. ReDMark(Ahmadi et al. 2020) extends this approach by adopting a residual structure for the encoder and adjusting the intensity of the watermark pattern by an intensity factor. StegaStamp (Tancik, Mildenhall, and Ng 2020) focuses on print-shooting robustness, simulating the print-shooting process with several differentiable operations within the noise layer. The idea of distortion simulation has influenced the following researches. RIHOOP (Jia et al. 2020) adds a differentiable distortion network to simulate the distortion caused by camera imaging during the training process so that the watermark information is not affected by the camera. Another research (Li, Liao, and Wu 2024) goes a step further by considering the effect of grayscale deviation on the screen in the cross-media screen-camera process, thus constructing a more realistic distortion layer to improve the model performance.

All of the above approaches maintain the robustness of the model by constructing a differentiable noise layer to approximate the perturbations of real-world images. This approach requires accurate modelling of the distortion, which is often difficult to achieve for complex distortions in the real world. Therefore, this approach is not generalisable.

Image watermarking free from differentiable noise layer. A number of approaches have been proposed to address the limitations of the differentiable noise layer. TDSL (Liu et al. 2019) separates the training process of the encoder and the decoder, thereby eliminating the differentiability constraint of the noise layer. In TDSL, the model is initially trained jointly without noise in the first stage, followed by training the decoder alone under real noise. However, this two-stage training strategy has an obvious problem: the encoder and decoder can only be optimized to their respective optimal solutions, but not to the global optimal solution, and it is difficult to ensure that the encoder will not be overfitted during the one-stage training process, thus resulting in the information being completely corrupted after the distortion. For this reason Forward ASL (Zhang et al. 2021) reintegrates the training process into an end-to-end architecture and treats all noise as additive noise, and considers only lossless watermarked images in the backpropagation without considering the effect of noise on the gradient. This approach is quite clean and simple, but since it does not consider the effect of noise on the backpropagation, this makes the gradient information obtained by the encoder may be too different from the true gradient information, which results in performance degradation.

TDSL's two-stage training strategy results in insufficient joint training between the encoder and decoder, and Forward ASL's approach oversimplifies by treating arbitrary distortions as additive noise. Both methods fail to effectively incorporate real non-differentiable distortions into end-to-end training. In our **Experiments**, we demonstrate that these limitations hinder both methods' ability to resist specific types of distortion. For this reason, we need to find a new way to to jointly train the encoder and decoder end-to-end, while at the same time making the Decoder robust enough to various non-differentiable distortions. In addition, We hope that this approach is effective and general enough to be compatible with most watermarking frameworks.

Methodology

Preliminary

Before delving into the specific details of our method, we first outline the basic assumptions and premises underlying our approach. DNN-based Image Watermarking often rely on differentiable approximations of distortions, which limits their effectiveness against real-world non-differentiable distortions. Therefore, it is essential to develop a watermarking solution capable of effectively handling any form of non-differentiable distortion. The effectiveness of differentiable noise layers is typically attributed to data enhancement, with Forward ASL(Zhang et al. 2021) showing that distortions primarily impact forward propagation. Based on this, we suggest that the Encoder can bypass the noise layer and adaptively refine the embedding strategy by receiving gradients from the robust Decoder alone to solve the non-differentiable problem. In our experiments, we demonstrate this by analyzing residuals under various noise attacks. Therefore, the key challenge lies in effectively training a robust decoder. Inspired by self-supervised contrastive learning (Grill et al. 2020; Caron et al. 2021; Zbontar et al. 2021), we align the feature vectors of distorted and clean watermarked images. This alignment ensures that if the clean image's features are correctly decoded, the distorted image's features will also be decoded correctly. Unlike self-supervised learning, our approach has a clear downstream goal (extracting watermark information), so we avoid issues with intermediate features collapsing into constant values and losing semantic information.

Regular Pipline

First we illustrate the pipeline of the general framework, which typically follows an END architecture in standard watermarking networks.

As shown in Figure 1 (a), it contains three main modules: Image Encoder f_θ, Message Decoder g_δ, and differentiable noise layer ϕ, where θ, δ are the parameters of the encoder and decoder, respectively. Given an image set $\mathcal{D}$, an image $x \in \mathbb{R}^{3\times h\times w}$ sampled uniformly from $\mathcal{D}$, and a binary watermark message $m \in \{0,1\}^n$ of length n. The watermarked image is denoted as $\hat{x}$. The image after noise attack is $\tilde{x}$, and the extracted watermark information by the decoder is $\hat{m}$. the watermark embedding process can be described as follows:

$$\hat{x} = f_\theta(x, m) \tag{1}$$

The simulation of a noise attack can be represented as follows:

$$\tilde{x} = \phi(\hat{x}) \tag{2}$$

The process of decoding the watermark information can be represented as:

$$\hat{m} = g_\delta(\tilde{x}) \tag{3}$$

Our goal is to simultaneously optimize θ, δ such that the following equation holds:

$$\arg\min_{\theta,\delta} \mathrm{E}_{x\sim\mathcal{D},m\sim\{0,1\}^n} \left[\|\hat{x} - x\| + \lambda \|\hat{m} - m\|\right] \tag{4}$$

where λ is the hyperparameter used to balance visual quality and decoding accuracy. From the above equation, it can be seen that $\tilde{x}$ is very critical for g_δ. Because if $\tilde{x}$ itself does not carry information m, the decoder is unlikely to be better than a random selection no matter how much δ is optimised. Therefore $\lambda \|\hat{m} - m\|$ has to constrain the optimisation of not only δ but also θ, which requires that the noise layer ϕ has to be differentiable, allowing the gradient to propagate from the decoder to the encoder. This limitation results in deep learning based watermarking frameworks not being able to maintain good robustness in the face of black-box noise and non-differentiable noise.

END^2 Pipline

Our pipline is shown in Figure 1 (b). To illustrate, we split the decoder in more detail. Let $g_\delta(\tilde{x}) = \gamma(\xi(\tilde{x}))$, $z \triangleq \xi(\tilde{x})$, where $\xi(\cdot)$ represents the feature extraction of the input image $\tilde{x}$ to obtain the feature vector $z \in \mathbb{R}^d$. The d stands for the dimension of the latent space. The $\gamma(\cdot)$ then represents the prediction of the final watermark information based on the feature vector, which consists of a linear layer. In contrast to the regular pipline we employ two decoders with the same structure but different roles in the training process, called Teacher Decoder **TD** and Student Decoder **SD**. Since they have the same structure, we use g_t and g_s to differentiate them in the following.

Our proposed method incorporates three additional components beyond the standard framework: (1) a dual-decoder for information extraction, (2) a feature alignment loss $\mathcal{L}_{s,t}$ into the primary loss function $\mathcal{L}$, and (3) the swapping learning and momentum updating strategy.

TD receives clean watermarked images to extract information and performs end-to-end optimization to ensure high quality watermark embedding and extraction. Instead SD receives the watermarked image with non-differentiable distortion directly. It serves to imitate the TD by aligning the feature vectors in the latent space, making the model robust to various distortions.

In addition, we have a projection layer φ shared by TD and SD, whose role is to project the feature vectors of both decoders into the same high-dimensional space. It is implemented by a bias-free linear layer. And the non-differentiable noise layer ψ, which is used to implement various non-differentiable distortion operations. During experiments, in order to ensure its non-differentiable nature, we use the stop gradient technique $sg[\cdot]$ to simulate the non-differentiable process.

The algorithm 1 shows the training flow of our method. Compared to the general END model process, our method adds one more Student Decoder and introduces momentum updating and swapping learning after training step. Furthermore, our approach remains compatible with differentiable noise layers without modification, enabling seamless integration with all END models.

Feature Alignment in Latent Space

In order for SD to learn how to deal robustly with a variety of non-differentiable distortions, we project the intermediate features of the two decoders onto a hypersphere and minimize the angle between their feature vectors.

Algorithm 1: END^2 Pipeline

Require: Image-Message pairs (x, m) from dataset, Encoder f_θ, Teacher Decoder g_t and Parameters δ_t, Student Decoder g_s and Parameters δ_s, Distortion Function ψ, Momentum Coefficient τ, Optimizer op, Swapping interval k

Ensure: Optimized Encoder and Decoders

for $i, (x, m)$ **in** enumerate(dataset) **do**
  $\hat{x} \leftarrow f_\theta(x, m)$ # Encode image with watermark
  $\tilde{x} \leftarrow sg[\psi(\hat{x})]$ # Apply distortion attack

  # extract message and feature from encoded images
  $\hat{m}_t, z_t \leftarrow g_t(\hat{x})$ # Teacher decoder
  $\hat{m}_s, z_s \leftarrow g_s(\tilde{x})$ # Student decoder

  $\mathcal{L} \leftarrow \mathcal{L}_{s,t} + \mathcal{L}_{msg} + \mathcal{L}_{quality}$ # Calculate total loss
  $\mathcal{L}$.backward() # Backpropagate
  op.step() # Update model parameters

  # Momentum updating strategy
  $\delta_t \leftarrow \tau\delta_t + (1-\tau)\delta_s$
  # Swapping Learning strategy
  if $(i+1) \mod k = 0$ **then**
    $g_t, g_s \leftarrow g_s, g_t$
  end if
end for

First extract the feature vectors:

$$\begin{aligned} z_t &= \xi_t(\hat{x}) \\ z_s &= \xi_s(sg[\psi(\hat{x})]) \end{aligned} \tag{5}$$

The $sg[\cdot]$ here is only added to ensure that the differentiable distortion also maintains its non-differentiable character in experiments, and can be removed in deployment.

Afterwards we project both feature vectors onto the same hypersphere via the projection layer φ:

$$\begin{aligned} \bar{z}_t &= \frac{\varphi(z_t)}{\|\varphi(z_t)\|_2} \\ \bar{z}_s &= \frac{\varphi(z_s)}{\|\varphi(z_s)\|_2} \end{aligned} \tag{6}$$

We also refer to the above operation as the projection operation $\mu(\cdot)$.

Finally we use cosine similarity to bring $\bar{z}_s$ closer to $\bar{z}_t$, which we call feature alignment loss:

$$\mathcal{L}_{s,t} = 2 - 2\frac{\langle \bar{z}_s, sg[\bar{z}_t] \rangle}{\|\bar{z}_s\|_2 \cdot \|sg[\bar{z}_t]\|_2} \triangleq \|\bar{z}_s - st[\bar{z}_t]\|_2^2 \tag{7}$$

Here we only let z_s be close to z_t, hence the need for gradient truncation function $sg[\cdot]$. Considering it from another perspective, we are essentially clustering in the latent space of watermarked information, and $\bar{z}_t$ is the centre of our clustering, and we keep all the feature vectors of the noisy image with the same information close to this centre.

Total Loss Function

Apart from the the feature alignment loss $\mathcal{L}_{s,t}$, our model includes the message loss $\mathcal{L}_{msg}$, aimed at ensuring the model can effectively extract the embedded information, and the quality loss $\mathcal{L}_{quality}$, which is essential for maintaining the visual quality of the watermarked images.

The message loss can be expressed by the following equation:

$$\mathcal{L}_{msg} = \|\hat{m}_t - m\|_2 + \|\hat{m}_s - m\|_2 \tag{8}$$

The quality loss can be represented as follows:

$$\mathcal{L}_{quality} = \|\hat{x} - x\|_2 \tag{9}$$

The total loss can be expressed as:

$$\mathcal{L} = \lambda_1\mathcal{L}_{s,t} + \lambda_2\mathcal{L}_{msg} + \lambda_3\mathcal{L}_{quality} \tag{10}$$

where $\lambda_1, \lambda_2, \lambda_3$ are three hyperparameters used to balance different losses, defaulting to $0.01, 8, 5$.

Swapping Learning and Momentum Updating

Since the TD only processes clean watermarked images, it cannot directly learn a decoding method robust to distortions. Consequently, the feature alignment loss will let the SD to emulate TD's representation ability, potentially leading to performance degradation. To ensure the consistency of representational capabilities between the two decoders, we employ swapping learning and momentum updating strategies. These approaches effectively enhance the robustness of our model.

Swapping Learning. To ensure that the TD experiences distortions, we introduce a straightforward approach: swapping the two decoders after each k training batches, with k defaulting to 1. What's more, the feature alignment loss ensures that, after the swap, the encoded features of the original TD can close to those of the original SD. This mutual supervision between the decoders facilitates the generation of consistent representations.

Momentum Updating. This strategy enables the TD to learn from the non-differentiable noise processed by the SD, thereby producing features better suited for decoding noisy images. Additionally, constant momentum updating helps ensure that both decoders gradually converge to similar parameters, maintaining consistent decoding capabilities. The process is illustrated as follows:

$$\begin{aligned} \delta_s &\longleftarrow \text{optimizer}(\delta_\text{s}, \nabla\mathcal{L}_\text{s,t}, \eta) \\ \delta_t &\longleftarrow \tau\delta_t + (1-\tau)\delta_s \end{aligned} \tag{11}$$

where τ is the weight factor and η is the learning rate, defaulting to 0.999 and $8e^{-4}$.

In Section **Ablation Study**, we will show that both strategies are effective in improving the performance of the model. And when the two are combined, they can produce even better results.

Experiments

Implementation Details

Our primary interest is to explore the performance of END^2 under fully non-differentiable distortion. In addition we also

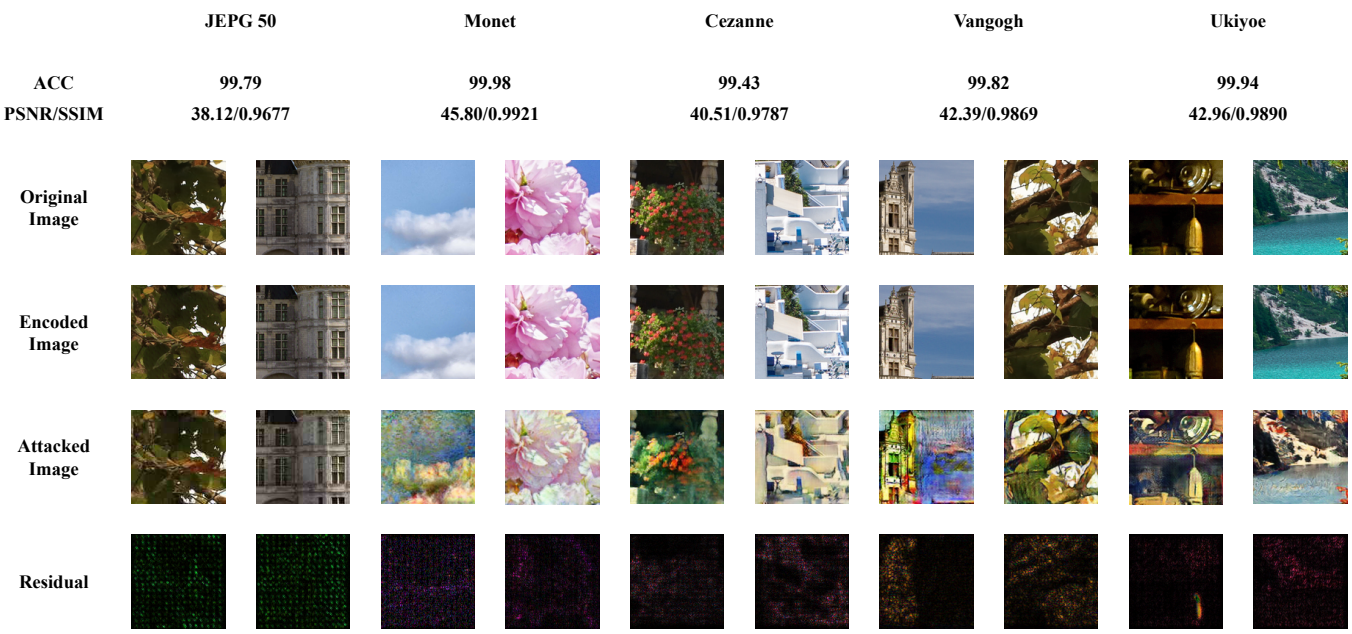

Figure 2: Results of the invisibility and robustness of our model under real JPEG compression and four style transfer distortions. The second row depicts the original image, while the third row shows the embedded watermarked image. The fourth row illustrates the watermarked image after being subjected to non-differentiable distortion. The final row shows the residual image, which is the difference between watermarked images and original images, magnified by a factor of 10 for enhanced visibility.

explore whether our approach is more competitive compared to other watermarking models that require a differential noise layer. Since our approach is a generic framework, we do not concern ourselves with the specific implementations of the Encoder and Decoder. Therefore, we employ the model structure in MBRS (Jia, Fang, and Zhang 2021). The model is trained on DIV2K(Agustsson and Timofte 2017). Specifically, we randomly select a block of size 128×128 from the training set as the cover image, and randomly sample a bit stream of length 30 to use as watermarking information for embedding. To evaluate the model's generalization, we also randomly select 2000 images from the COCO (Lin et al. 2014) and ImageNet (Deng et al. 2009) datasets for testing. We utilize the Adam optimizer with a fixed learning rate of $8e^{-4}$ for training. The batch size is set to 32, and the model is trained for a total of 5000 epochs on a NVIDIA RTX 3090 GPU.

Metrics. We use average bit accuracy (ACC) to evaluate the decoding accuracy of the model. For visual quality assessment, we use the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) as metrics.

Baselines. We compare against TDSL (Liu et al. 2019) and Forward ASL (Zhang et al. 2021) for non-differentiable distortion. Additionally, to demonstrate the superiority of our method, we compare it with three classical models HiDDeN(Zhu et al. 2018), StegaStamp(Tancik, Mildenhall, and Ng 2020) and MBRS (Jia, Fang, and Zhang 2021) which need differentiable Noiser.

	END^2	TDSL	Forward ASL	MBRS	HiDDeN	StegaStamp
Non-Diff Noiser	✓	✓	✓			
ACC	**94.55**	94.10	92.37	89.17	87.86	88.90
PSNR	**45.62**	36.32	33.48	40.64	32.62	35.34
SSIM	**0.9897**	0.9726	0.9054	0.9796	0.8486	0.8810

Table 1: Average decoding accuracy and visual quality of the different methods under the random distortion.

Comparison with Previous Methods

In this section, we compare our method with two SOTA models (Liu et al. 2019; Zhang et al. 2021) that also address non-differentiable distortion. At the same time we choose two classical models (Zhu et al. 2018; Tancik, Mildenhall, and Ng 2020) that require a differentiable noise layer to illustrate that even without the need for a differentiable noise layer our approach is still competitive. In addition, since our model maintains the same structure as MBRS (Jia, Fang, and Zhang 2021), we trained the MBRS model under differentiable conditions for comparison.

We use a combined noise to simulate complex distortion conditions in the real world. The noise layer consists of Roate, Crop, Translate, Scale, Shear, Dropout, Cropout, Color transformation, JPEG compression, Gaussian Filter, Gaussian Noise.

To quantitatively analyze the relationship between visual quality and decoding accuracy, we measure the performance of different models under random noise attacks. Table 1 shows the performance under random distortion. Our ap-

Dataset	Method	Identity	Gaussian Filter ($\sigma = 2$)	JPEG ($Q = 50$)	Crop ($p = 0.1$)	Dropout ($p = 0.5$)	Rotate ($deg = 10$)	Translate ($dis = 0.05$)	Scale ($f = 0.65$)	Gaussian Noise ($std = 0.01$)
DIV2K	Proposed	**100**	**99.46**	**95.67**	93.21	**100**	98.26	**99.76**	**94.37**	**100**
	TDSL	98.92	60.18	55.21	64.87	99.56	**99.85**	81.67	88.71	98.89
	Forward ASL	99.99	83.68	90.21	92.43	99.23	92.48	89.38	70.63	99.99
	MBRS	98.16	95.31	86.43	**94.24**	98.59	95.22	94.70	91.78	98.14
	HiDDeN	89.56	59.66	53.45	87.04	71.50	82.62	82.65	68.59	88.31
	StegaStamp	90.94	89.92	84.55	88.61	77.53	86.60	86.54	84.10	90.76
COCO	Proposed	**100**	**99.22**	**94.89**	**93.08**	**99.99**	98.21	**99.77**	**93.77**	**100**
	TDSL	99.77	58.32	53.94	63.52	96.72	**99.75**	80.36	86.33	99.63
	Forward ASL	99.98	85.57	92.05	88.33	98.46	92.02	88.98	69.74	99.94
	MBRS	96.99	94.92	90.59	90.74	90.39	90.79	91.21	91.01	91.69
	HiDDeN	86.26	73.11	66.49	70.92	70.44	71.87	72.76	72.00	73.50
	StegaStamp	89.58	89.33	87.56	87.41	84.91	85.18	85.20	85.01	85.59
ImageNet	Proposed	**100**	**99.33**	**93.71**	**94.13**	**99.99**	**98.33**	**99.79**	**93.68**	**100**
	TDSL	99.59	59.73	54.06	64.60	97.70	98.28	81.20	86.59	99.38
	Forward ASL	99.94	86.83	91.94	89.08	98.64	91.11	88.40	69.70	99.97
	MBRS	96.42	93.85	89.59	90.00	95.19	90.19	90.67	90.36	90.97
	HiDDeN	86.24	72.54	66.17	70.49	69.97	71.37	72.21	71.35	72.80
	StegaStamp	88.99	88.33	86.63	86.23	83.71	83.89	83.90	83.49	84.00

Table 2: Benchmark comparisons on robustness against different distortions. We trained the models only on DIV2K and tested them on three different datasets. By adjusting the embedding strength, the visual quality of all models was maintained at PSNR=35.

proach, TDSL and Forward ASL do not require a differentiable noise layer, so we truncate the gradient of the distorted image in the actual training. Our method significantly outperforms the baselines. Although TDSL achieves similar decoding accuracy, its redundant embedding strategy results in a PSNR nearly 10 points lower than ours. Notably, our approach even surpasses MBRS trained with a differentiable noise layer, demonstrating that such a layer is non-essential.

For a comprehensive evaluation, we test the decoding accuracy of our trained model across three datasets using a variety of distortions: Identity, Gaussian Filter ($\sigma = 2$), JPEG compression ($Q = 50$), Crop ($p = 0.1$), Dropout ($p = 0.5$), Rotate ($deg = 10$), Translate ($dis = 0.05$), Scale ($f = 0.65$) and Gaussian Noise($std = 0.01$). Table 2 presents the experimental results. Our method maintains a decoding accuracy of over 90% under all noise attacks, demonstrating its robustness against a wide range of distortions. TDSL struggles to resist Gaussian Filter, JPEG Compression, and Crop attacks, while Forward ASL performs poorly under Scale attacks. This suggests that both methods are less robust to certain types of distortions, despite their claims of handling arbitrary distortions.

Performance under Non-Differentiable Distortion

To further demonstrate the robustness of our method under real non-differentiable distortions, we select real JPEG Compression and four style transfer black-box distortions for testing. For JPEG compression, we utilized the PIL package in Python, while the four style transfer distortions are generated by a pre-trained CycleGAN (Zhu et al. 2017) model.

Real JPEG Compression. We train our model using only real JPEG Compression with a quality factor of $Q = 50$ and then test it under various quality factors. To be fair, we fix the PSNR at 38. Notably, the model trained with the combined noise layer in the previous subsection is expected to demonstrate some level of generalization performance. Therefore, we consider it as a baseline to evaluate its robustness against real-world black-box noise, thereby underscoring the necessity of training against black-box distortions.

	Real JPEG Compression				
	50	40	30	20	10
Proposed	**99.95**	94.39	**83.41**	**68.22**	**55.20**
Pre-trained	69.02	64.45	60.53	55.58	51.49
TDSL	74.58	70.05	63.50	57.32	52.19
Forward ASL	99.76	**96.39**	80.43	60.83	51.41

Table 3: Result of real JPEG compression.

Table 3 presents the results of our experiments. Our proposed method consistently outperforms most baseline models, proving the validity of our approach. Moreover, it is evident that the model pre-trained with a combined noise layer does not perform well on real JPEG compression, even though simulated JPEG compression was included in the training process. This discrepancy indicates a significant gap between the simulated noise layer and real-world noise. Therefore, it is crucial to develop a method that can robustly handle arbitrary black-box distortions encountered in real-world scenarios.

Black-box Style Transfer Distortions. Testing with style transfer distortions is essential. Because it simulates complex, non-differentiable transformations that occur in the real world. Unlike traditional noise or compression, style transfer introduces variations in texture, color, and structure, challenging the model with intricate distortions. To validate

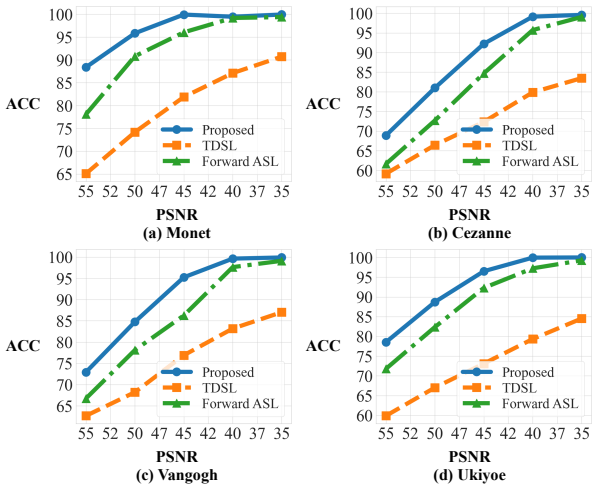

Figure 3: The performance of different models, in terms of average decoding accuracy, under four style transfer distortions is evaluated as PSNR varies.

our method, we selected four styles of migration distortions (Monet, Cezanne, Vangogh and Ukiyoe) .

As illustrated in Figure 2, our method maintains a decoding accuracy above 99% across various distortions, while also preserving high visual quality. Furthermore, to illustrate that our approach allows the Encoder to perceive different distortions, we draw the JPEG Compression with quality factor $Q = 50$ also in Figure 2. The residual images reveal that the encoder adopts distinct encoding strategies for different types of distortions. This indicates that our approach enables the Encoder to recognize patterns in non-differentiable distortions and apply a more robust coding strategy.

In addition, we evaluated the robustness of each model against the four style transfer distortions by varying the watermark embedding strength. As shown in Figure 3, our method consistently outperforms the baseline at various watermark strengths and achieves nearly 95% decoding accuracy at a PSNR of 45. These results demonstrate that our approach is robust to unknown black-box distortions.

Ablation Study

In this section, we conduct the ablation experiments to better illustrated the proposed architecture. We are primarily interest in the effectiveness of the momentum updating and swapping learning strategies, and how different feature alignment losses affect the performance of the model.

Importance of Different Compositions. Table 4 presents the ablation study results of our method under various configurations. The first row represents the outcomes without applying any specific strategies, while the subsequent rows illustrate the results obtained using different combinations of our proposed strategies. The experiments reveal that training without strategies results in a final decoding accuracy of approximately 78%. Introducing the feature alignment loss increases this accuracy by about 8%, demonstrating that aligning features in the latent space can effectively enhance the model's robustness. Furthermore, employing either the MU or SL strategies individually leads to additional improvements in robustness, albeit with a reduction in visual quality. When both strategies are applied simultaneously, the model achieves a peak decoding accuracy of approximately 95% while maintaining a PSNR of 45.

	ACC	PSNR	SSIM
None	78.93	45.04	0.9889
FA	86.79	**47.89**	**0.9921**
FA+MU	92.23	39.26	0.9620
FA+SL	93.96	43.68	0.9867
FA+MU+SL	**94.55**	45.62	0.9896

Table 4: The performance of the model under different structures. Where FA, MU, SL stand for feature alignment loss, momentum updating strategy and swapping learning strategy respectively.

	ACC	PSNR	SSIM
Proposed	94.55	**45.62**	**0.9896**
MSE	**94.70**	42.92	0.9837
DINO	91.39	43.23	0.9855

Table 5: Impact of different feature alignment losses on model performance.

Impact of Feature Alignment Loss. Furthermore, we explored the effect of different feature alignment losses on model performance by comparing the Mean Squared Error (MSE) loss and the feature alignment method mentioned in DINO (Caron et al. 2021). Table 5 presents the results of our experiments. Experiments indicate that although the direct use of MSE loss achieves a comparable ACC to our method, it results in a visual quality drop of about 3 points. In contrast, using DINO leads to a decrease in ACC. However, our method consistently maintains performance, showing that the choice of alignment loss does not significantly impact the overall effectiveness.

Conclusion

In this paper, we presented a novel dual-decoder architecture (END^2) to address the challenges posed by non-differentiable distortions in deep learning-based image watermarking. By aligning feature vectors between a Teacher Decoder for lossless images and a Student Decoder for distorted images, our approach overcomes the limitations of traditional END architectures that rely on differentiable noise layers. Experiments show that our method outperforms state-of-the-art algorithms in various distortion scenarios and can be easily integrated into existing END frameworks. Our findings highlight the potential of END^2 as a robust solution for real-world image watermarking challenges.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 61972169,62372203 and 62302186, in part by the National key research and development program of China(2022YFB2601802), in part by the Major Scientific and Technological Project of Hubei Province (2022BAA046, 2022BAA042), in part by the Knowledge Innovation Program of Wuhan-Basic Research, in part by China Postdoctoral Science Foundation 2022M711251.

References

Agustsson, E.; and Timofte, R. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *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.

Bassia, P.; Pitas, I.; and Nikolaidis, N. 2001. Robust audio watermarking in the time domain. *IEEE Transactions on multimedia*, 3(2): 232–241.

Bi, N.; Sun, Q.; Huang, D.; Yang, Z.; and Huang, J. 2007. Robust image watermarking based on multiband wavelets and empirical mode decomposition. *IEEE Transactions on Image Processing*, 16(8): 1956–1966.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9650–9660.

Cox, I. J.; Kilian, J.; Leighton, F. T.; and Shamoon, T. 1997. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12): 1673–1687.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284.

Haitsma, J.; Van Der Veen, M.; Kalker, T.; and Bruekers, F. 2000. Audio watermarking for monitoring and copy protection. In *Proceedings of the 2000 ACM workshops on Multimedia*, 119–122.

Hernandez, J. R.; Amado, M.; and Perez-Gonzalez, F. 2000. DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE transactions on image processing*, 9(1): 55–68.

Hsu, C.-T.; and Wu, J.-L. 1999. Hidden digital watermarks in images. *IEEE Transactions on image processing*, 8(1): 58–68.

Jia, J.; Gao, Z.; Chen, K.; Hu, M.; Min, X.; Zhai, G.; and Yang, X. 2020. RIHOOP: Robust invisible hyperlinks in offline and online photographs. *IEEE Transactions on Cybernetics*, 52(7): 7094–7106.

Jia, Z.; Fang, H.; and Zhang, W. 2021. 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*, 41–49.

Kandi, H.; Mishra, D.; and Gorthi, S. R. S. 2017. Exploring the learning capabilities of convolutional neural networks for robust image watermarking. *Computers & Security*, 65: 247–268.

Langelaar, G. C.; Setyawan, I.; and Lagendijk, R. L. 2000. Watermarking digital image and video data. A state-of-the-art overview. *IEEE Signal processing magazine*, 17(5): 20–46.

Li, Y.; Liao, X.; and Wu, X. 2024. Screen-Shooting Resistant Watermarking with Grayscale Deviation Simulation. *IEEE Transactions on Multimedia*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, Y.; Guo, M.; Zhang, J.; Zhu, Y.; and Xie, X. 2019. A novel two-stage separable deep learning framework for practical blind watermarking. In *Proceedings of the 27th ACM International conference on multimedia*, 1509–1517.

Swanson, M. D.; Zhu, B.; Tewfik, A. H.; and Boney, L. 1998. Robust audio watermarking using perceptual masking. *Signal processing*, 66(3): 337–355.

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.

van Schyndel, R.; Tirkel, A.; and Osborne, C. 1994. A digital watermark. In *Proceedings of 1st International Conference on Image Processing*, volume 2, 86–90 vol.2.

Wang, Y.; and Pearmain, A. 2006. Blind MPEG-2 video watermarking robust against geometric attacks: a set of approaches in DCT domain. *IEEE Transactions on Image Processing*, 15(6): 1536–1543.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, 12310–12320. PMLR.

Zhang, C.; Karjauv, A.; Benz, P.; and Kweon, I. S. 2021. Towards robust deep hiding under non-differentiable distortions for practical blind watermarking. In *Proceedings of the 29th ACM international conference on multimedia*, 5158–5166.

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.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.