THE SELF-RE-WATERMARKING TRAP: FROM EXPLOIT TO RESILIENCE

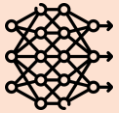
Vithurabiman Senthuran[1], [1] Yong Xiang, [1]Iynkaran Natgunanathan & [2]Uthayasanker Thayasivam
[1]Deakin University, [2]University of Moratuwa

Background

GOAL

Embed an invisible watermark into a cover image so that the watermarked image is identical to the cover image

DEEP LEARNING METHODS



Better imperceptibility and more robust to common image processing attacks

RESEARCH GAP

All the methods assume single-use embedding. None defends against malicious encoder re-use

Cross Model Re-Watermarking

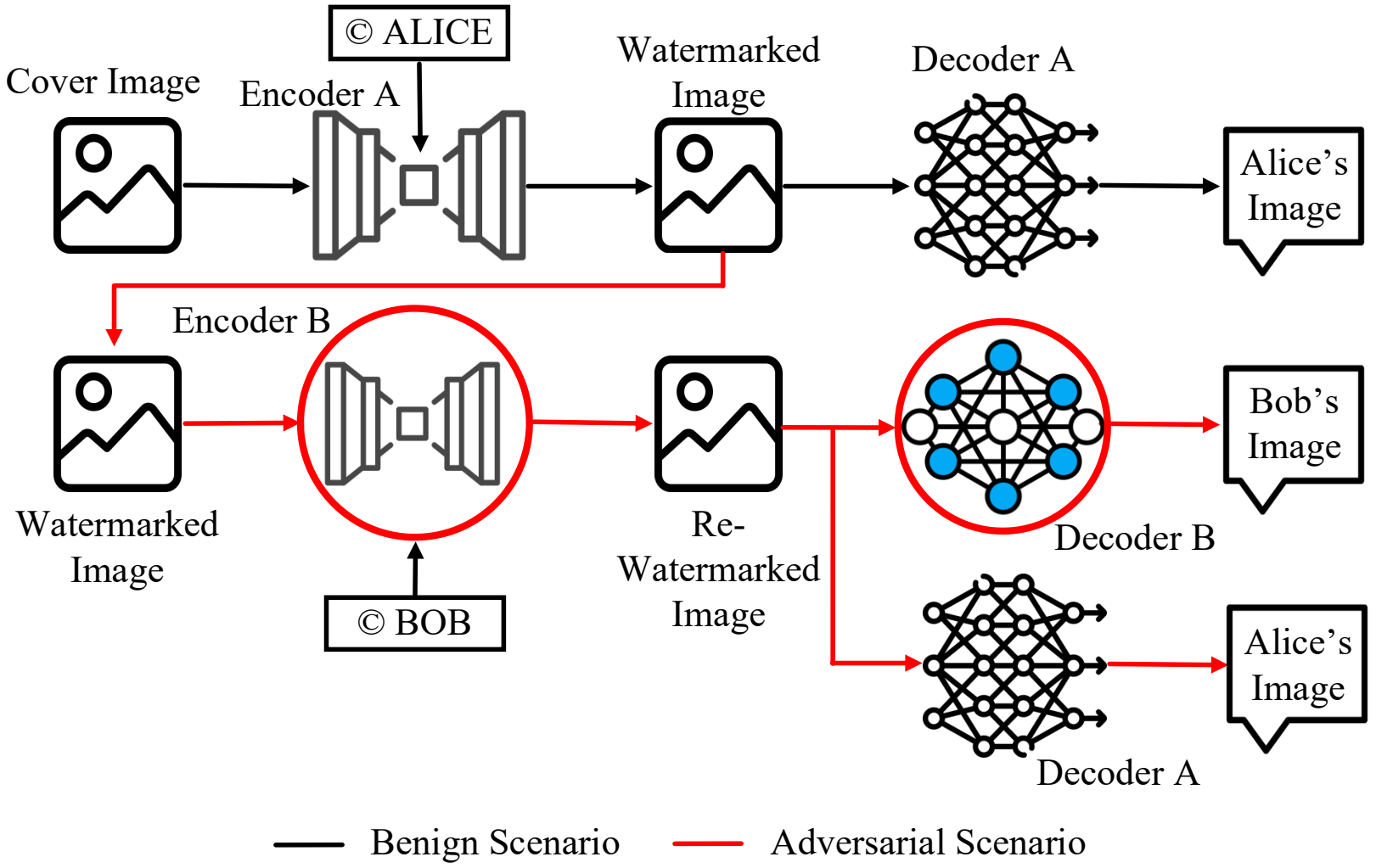


Figure 1: Cross Model Re-Watermarking

Self-Re-Watermarking

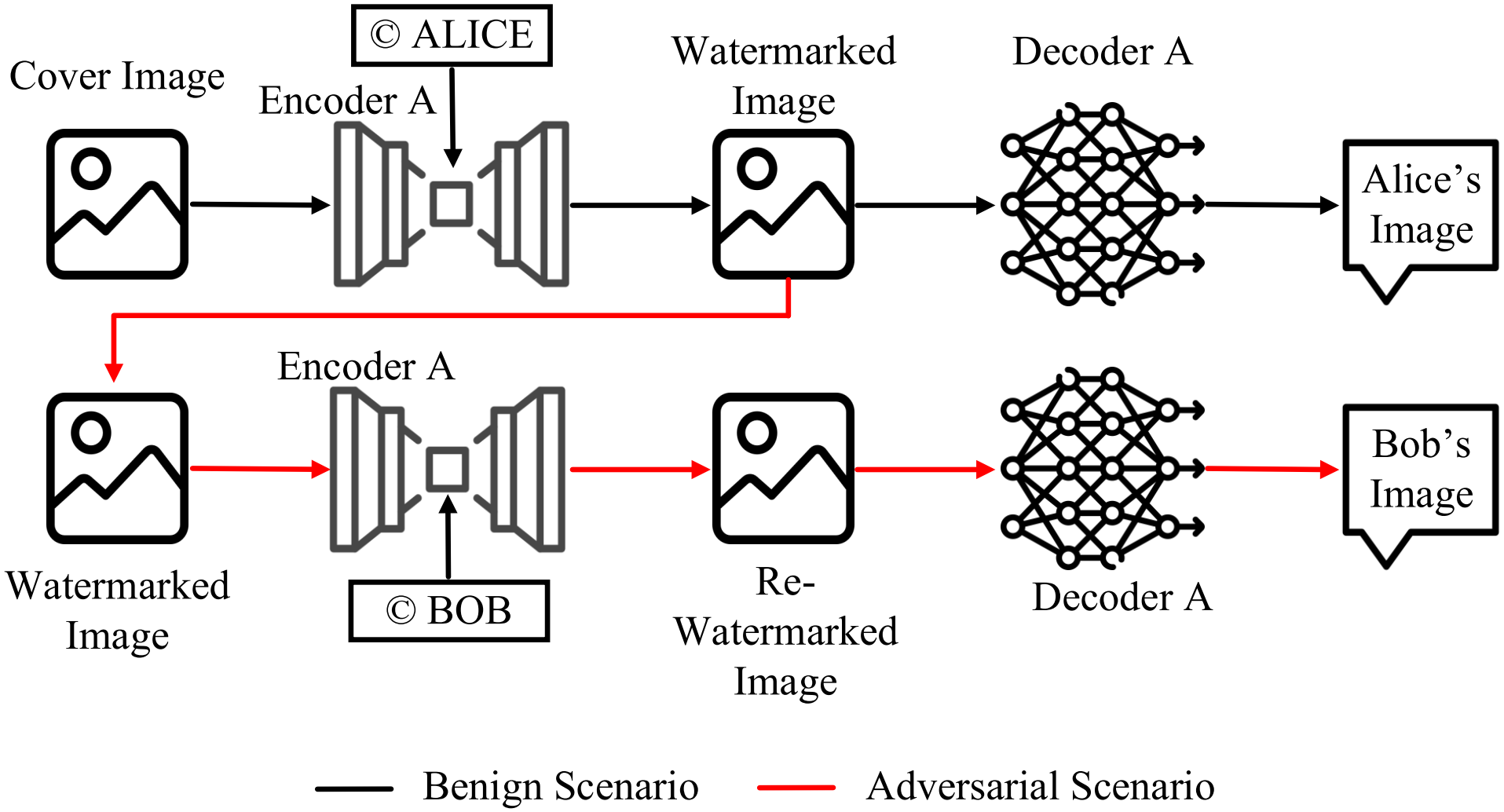


Figure 2: Self-Re-Watermarking

Solution

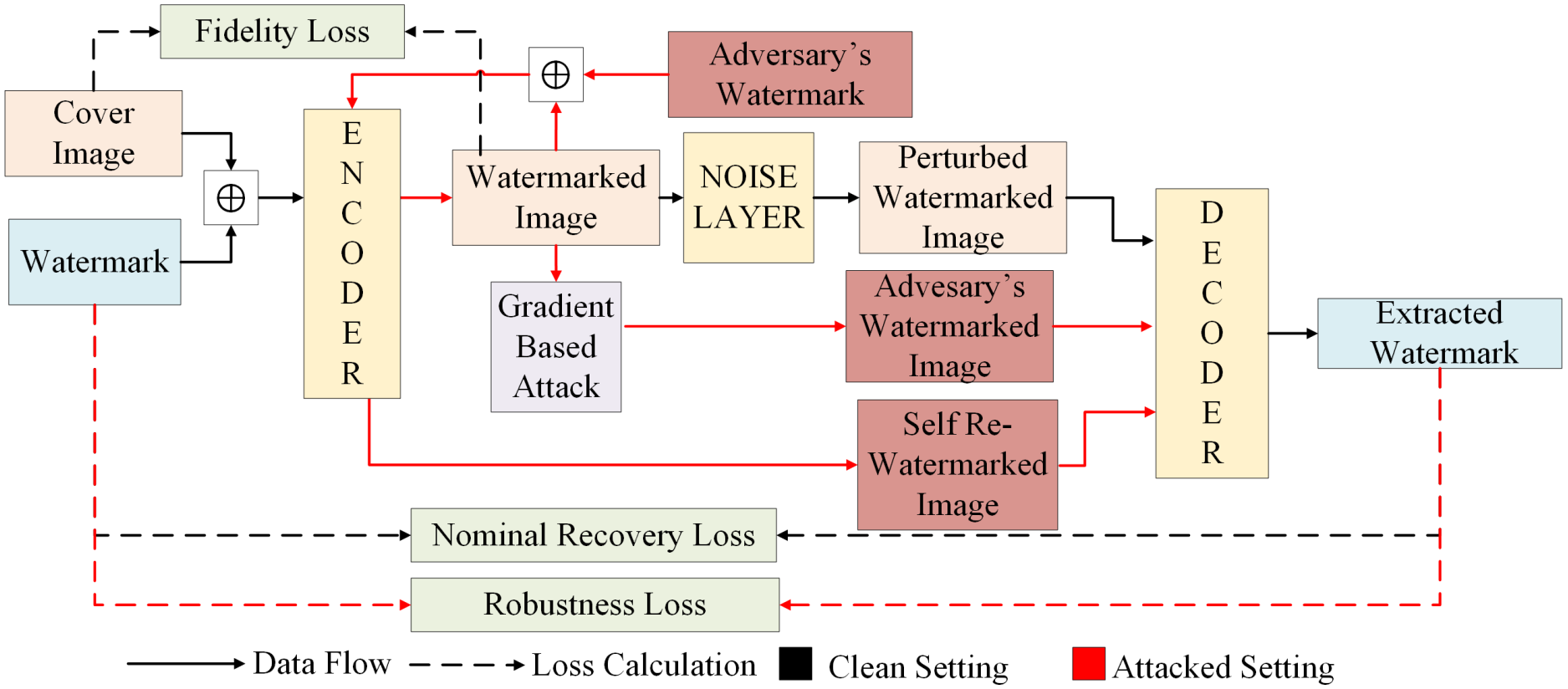


Figure 3: Proposed Solution

PGD-Based Overwriting Attack

Algorithm 1 PGD Self-Overwrite Attack

Require:

Watermarked image $x_w \in [-1,1]^{B\times C\times H\times W}$, Target message bits $m_g \in \{0,1\}^{B\times L}$, Decoder function D, Maximum perturbation ϵ, Step size α, Number of iterations T

Ensure: Adversarial image x_{adv}

Initialize $x_{adv} \leftarrow x_w$

for $i = 1$ **to** T **do**

    Compute logits: $Z \leftarrow D(x_{adv})$

    Compute loss: $\mathcal{L} = \text{BCEWithLogitsLoss}(Z, m_g)$

    Compute gradient: $g \leftarrow \nabla_{x_{adv}} \mathcal{L}$

    Gradient descent step with sign: $x_{adv} \leftarrow x_{adv} - \alpha \cdot \text{sign}(g)$

    Project perturbation back to ℓ_∞ ball: $\delta \leftarrow \text{clip}(x_{adv} - x_w, -\epsilon, \epsilon)$

    Clamp to valid image range: $x_{adv} \leftarrow \text{clip}(x_w + \delta, -1, 1)$

end for

return x_{adv}

Computing loss gradient to steer decoder output towards target message

PGD-Based Overwriting Attack

Algorithm 1 PGD Self-Overwrite Attack

Require:

Watermarked image $x_w \in [-1, 1]^{B \times C \times H \times W}$, Target message bits $m_g \in \{0, 1\}^{B \times L}$, Decoder function D, Maximum perturbation ϵ, Step size α, Number of iterations T

Ensure: Adversarial image x_{adv}

Initialize $x_{adv} \leftarrow x_w$

for $i = 1$ **to** T **do**

    Compute logits: $Z \leftarrow D(x_{adv})$

    Compute loss: $\mathcal{L} = \text{BCEWithLogitsLoss}(Z, m_g)$

    Compute gradient: $g \leftarrow \nabla_{x_{adv}} \mathcal{L}$

    Gradient descent step with sign: $x_{adv} \leftarrow x_{adv} - \alpha \cdot \text{sign}(g)$

    Project perturbation back to ℓ_∞ ball: $\delta \leftarrow \text{clip}(x_{adv} - x_w, -\epsilon, \epsilon)$

    Clamp to valid image range: $x_{adv} \leftarrow \text{clip}(x_w + \delta, -1, 1)$

end for

return x_{adv}

Gradient Descent

PGD-Based Overwriting Attack

Algorithm 1 PGD Self-Overwrite Attack

Require:

Watermarked image $x_w \in [-1,1]^{B\times C\times H\times W}$, Target message bits $m_g \in \{0,1\}^{B\times L}$, Decoder function D, Maximum perturbation ϵ, Step size α, Number of iterations T

Ensure: Adversarial image x_{adv}

Initialize $x_{adv} \leftarrow x_w$

for $i = 1$ **to** T **do**

  Compute logits: $Z \leftarrow D(x_{adv})$

  Compute loss: $\mathcal{L} = \text{BCEWithLogitsLoss}(Z, m_g)$

  Compute gradient: $g \leftarrow \nabla_{x_{adv}} \mathcal{L}$

  Gradient descent step with sign: $x_{adv} \leftarrow x_{adv} - \alpha \cdot \text{sign}(g)$

  Project perturbation back to ℓ_∞ ball: $\delta \leftarrow \text{clip}(x_{adv} - x_w, -\epsilon, \epsilon)$

  Clamp to valid image range: $x_{adv} \leftarrow \text{clip}(x_w + \delta, -1, 1)$

end for

return x_{adv}

Enforcing perturbation bounds and valid pixel range

Formal Analysis

- Given a triplet of cover image x, watermark m and adversarial watermark m', we can bound the bit error rate for the original watermark and the recovered watermark after the overwrite as

$$BER(x, m, m') \leq \frac{1}{L} \sum_{i=1}^{L} 1(\Delta_i(x, m) \leq K_D \delta_\infty) + \epsilon_{rec}$$

Length of the Watermark

Clean Margin of the Bit i

Lipschitz Constant of the Decoder

Perturbation under overwrite

Nominal Error

Experimental Results – Bit Recovery under Re-Watermarking

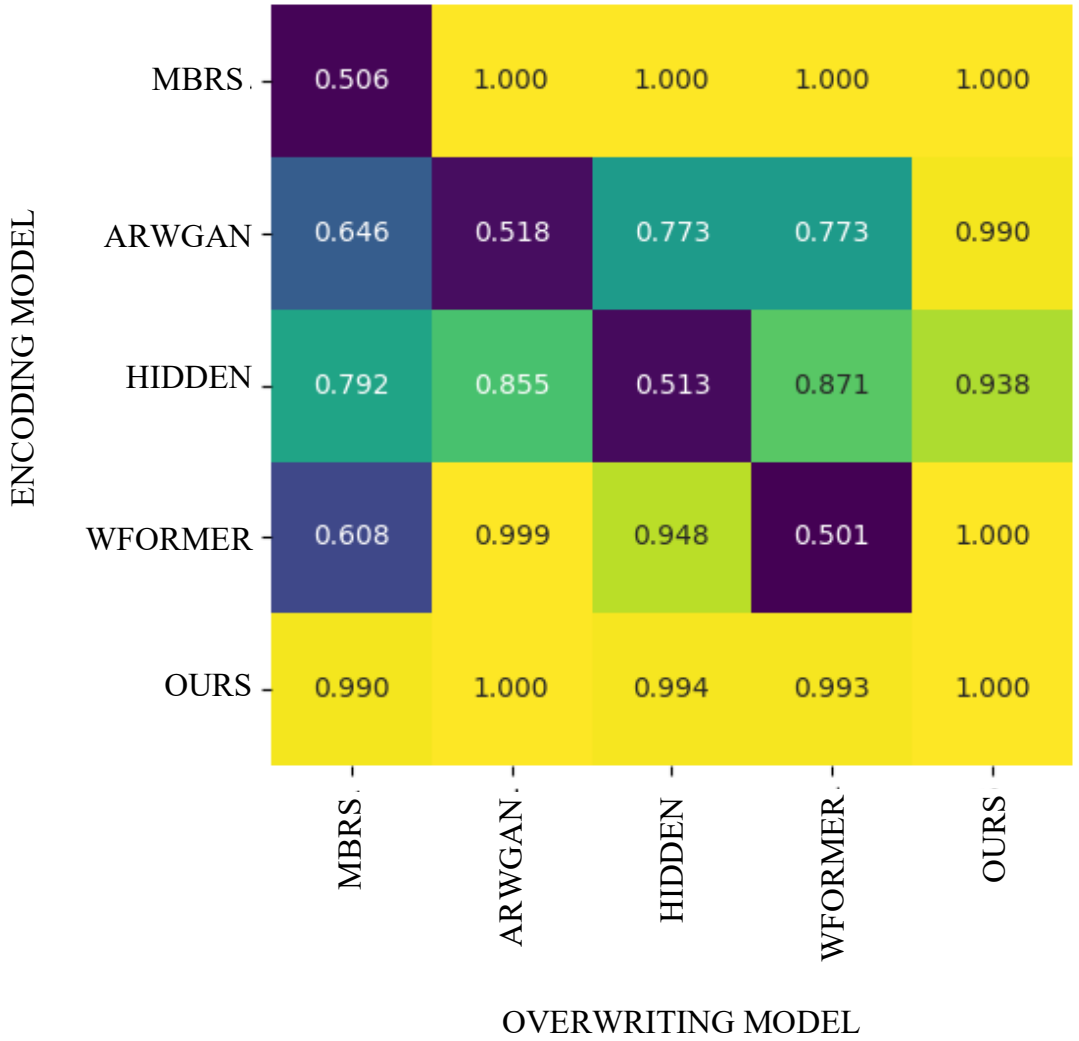


Figure 4: Bit Accuracy for decoded message with respect to original watermark under re-watermarked scenarios

Experimental Results- Encoder Reuse

- Robustness of the models under encoder-based re-watermarking

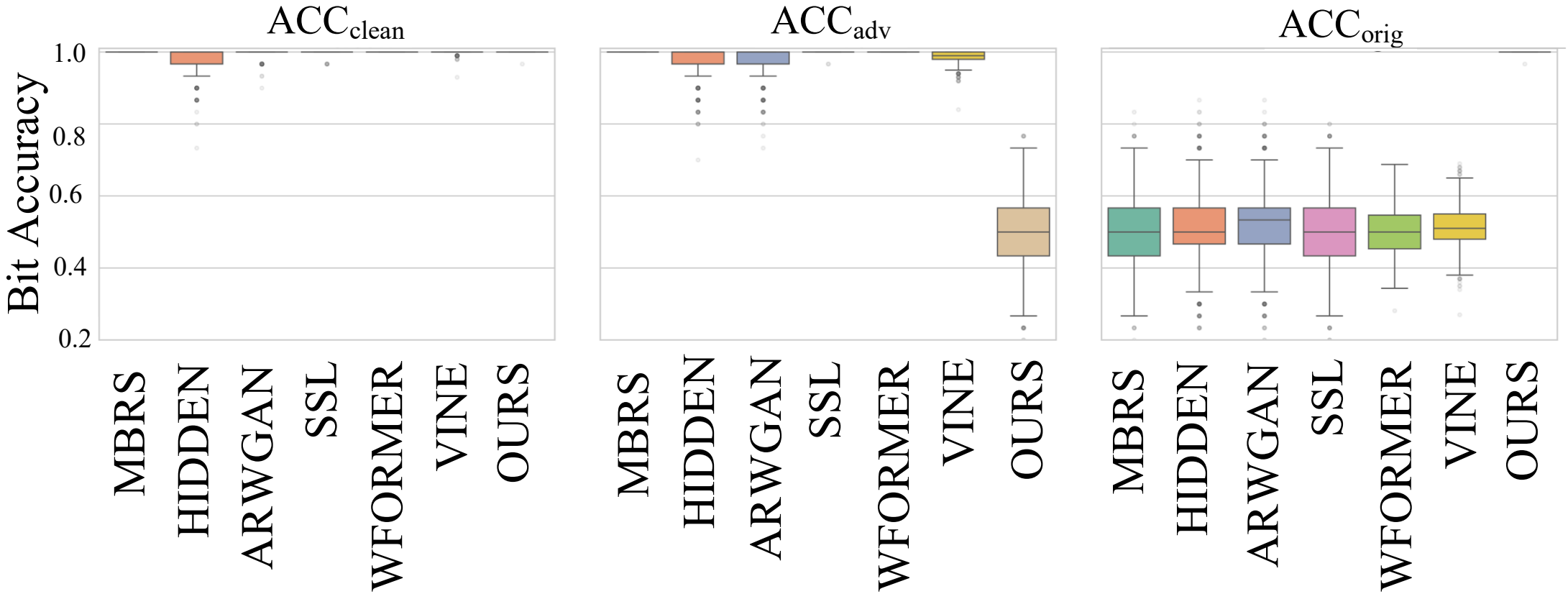


Figure 5: Bit accuracy under three scenarios: 1. Benign Scenario (ACC_{clean}), 2. Adversarial watermark (ACC_{adv}) after encoder based self-re-watermarking, and original watermark recovery after encoder based self-re-watermarking(ACC_{orig})

Experimental Results- PGD Attacks

- Experiments were performed with a perturbation budget $\epsilon = 0.03$, step size α=0.007 and 50 iterations

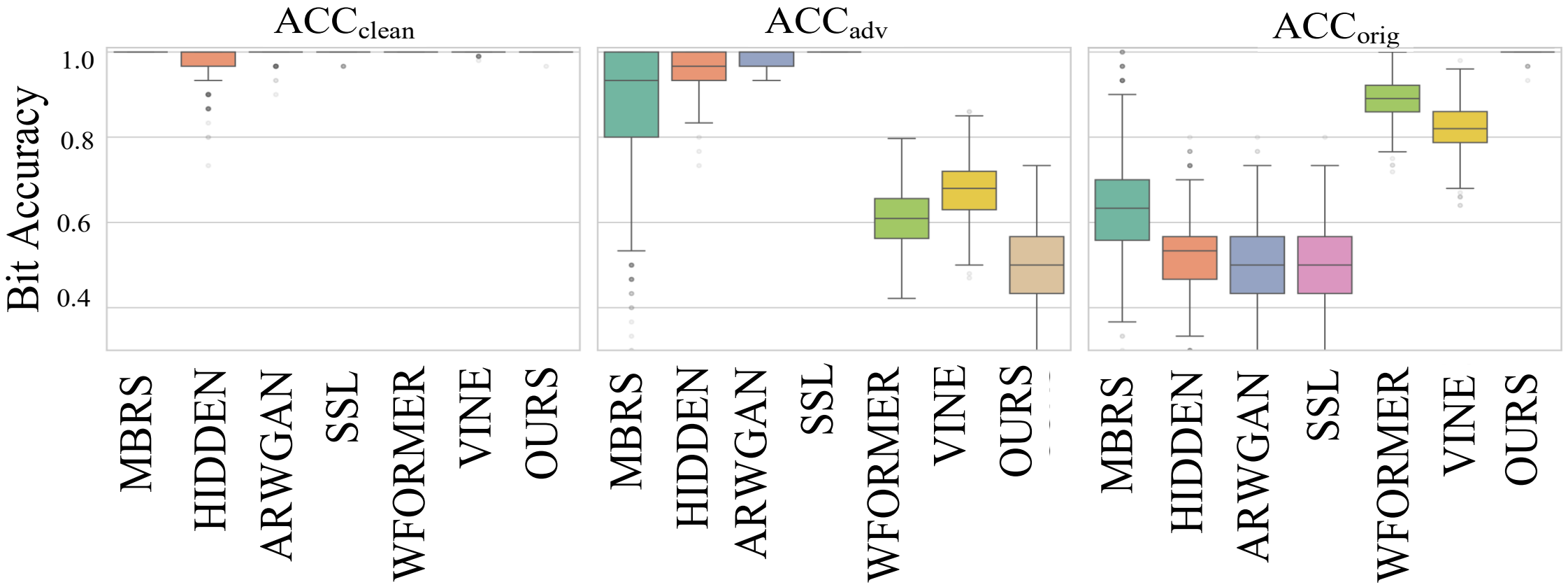


Figure 6: Bit accuracy under three scenarios: 1. Benign Scenario (ACC_{clean}), 2. Adversarial watermark (ACC_{adv}) after gradient based self-re-watermarking, and original watermark recovery after gradient based self-re-watermarking(ACC_{orig})

Experimental Results- PGD Attacks

- Experiments were performed with a perturbation budget $\epsilon = 0.04$, step size α=0.01 and 100 iterations

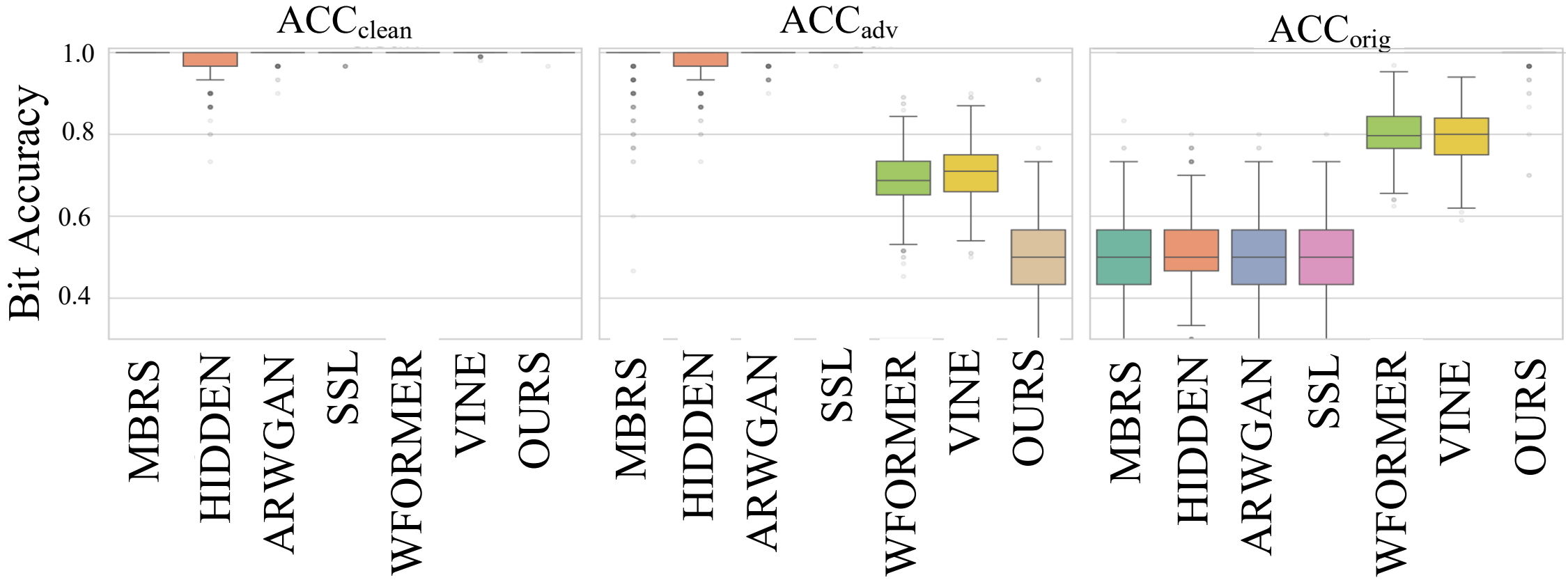


Figure 7: Bit accuracy under three scenarios: 1. Benign Scenario (ACC_{clean}), 2. Adversarial watermark (ACC_{adv}) after gradient based self-re-watermarking, and original watermark recovery after gradient based self-re-watermarking(ACC_{orig})

Experimental Results

Studies	Visual Quality		ACC_{clean} (%)					ACC_{orig} (%)		
	PSNR (dB)	SSIM	JPEG (50)	Gaussian Blur (2.0)	Dropout (30%)	Cropout (30%)	Crop (3.5%)	Self Re-embed	PGD Moderate	PGD Strong
dwtDctSvd	28.57	0.94	**99.97**	99.41	54.36	85.40	51.29	50.00	N/A	N/A
Hidden	33.55	0.92	63.00	96.00	93.00	94.00	88.00	51.29	52.03	51.45
MBRS	35.84	0.89	91.97	**100.00**	99.96	99.98	92.68	50.34	63.51	51.26
SSLW	33.10	0.94	83.01	98.96	88.11	79.66	50.73	49.90	49.81	49.81
ARWGAN	35.87	0.96	93.98	99.99	**100.00**	99.82	98.17	51.94	50.68	50.73
WFORMER	33.50	0.91	99.14	**100.00**	**100.00**	**100.00**	98.70	50.02	88.64	80.15
VINE	**37.07**	**0.99**	**99.97**	99.84	87.63	99.99	52.24	51.20	82.00	79.41
Proposed	34.03	0.97	95.06	99.66	98.90	98.14	**99.85**	**100.00**	**99.95**	**99.37**

Table 1: Robustness Evaluation on Image Processing and Adversarial Attacks

Robustness to Pixelwise attacks

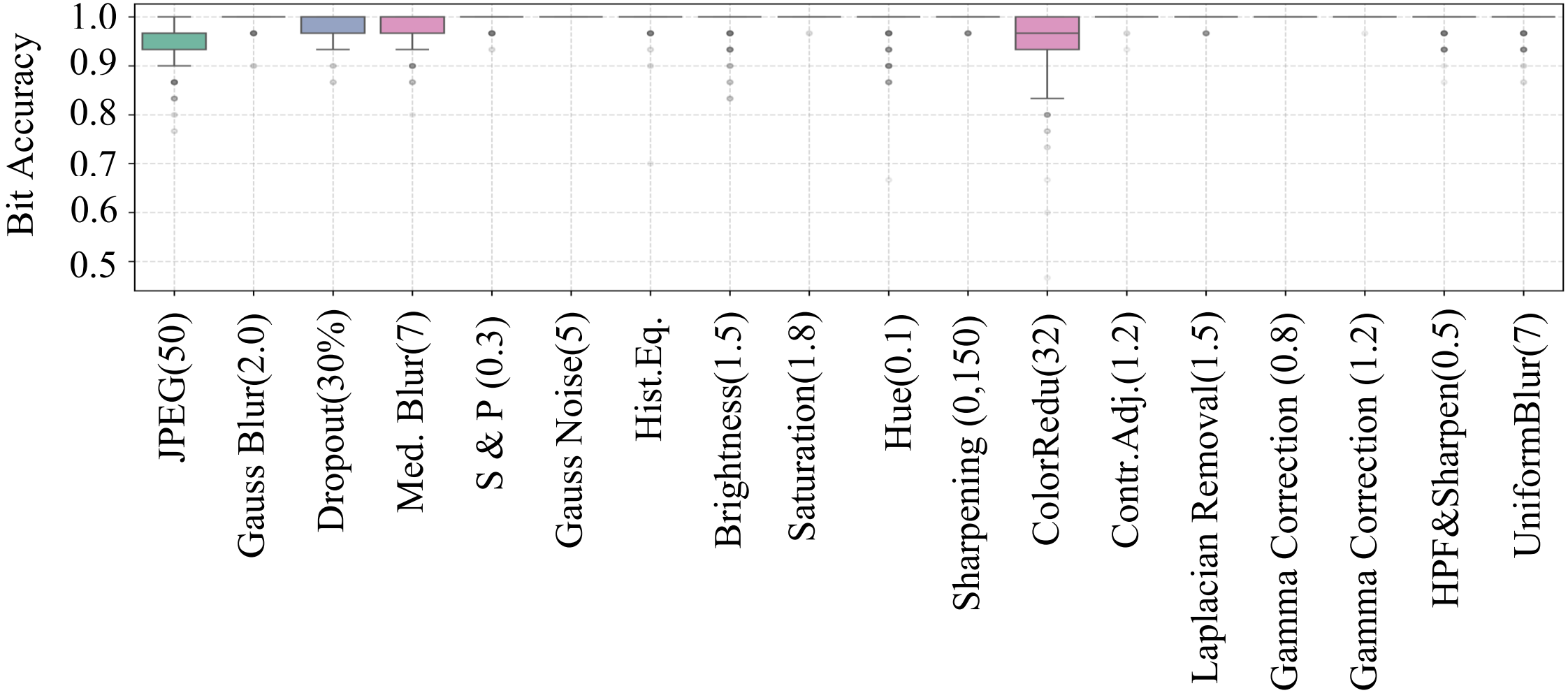


Figure 6: Bit accuracy under pixel-wise perturbations

Robustness to Geometric attacks

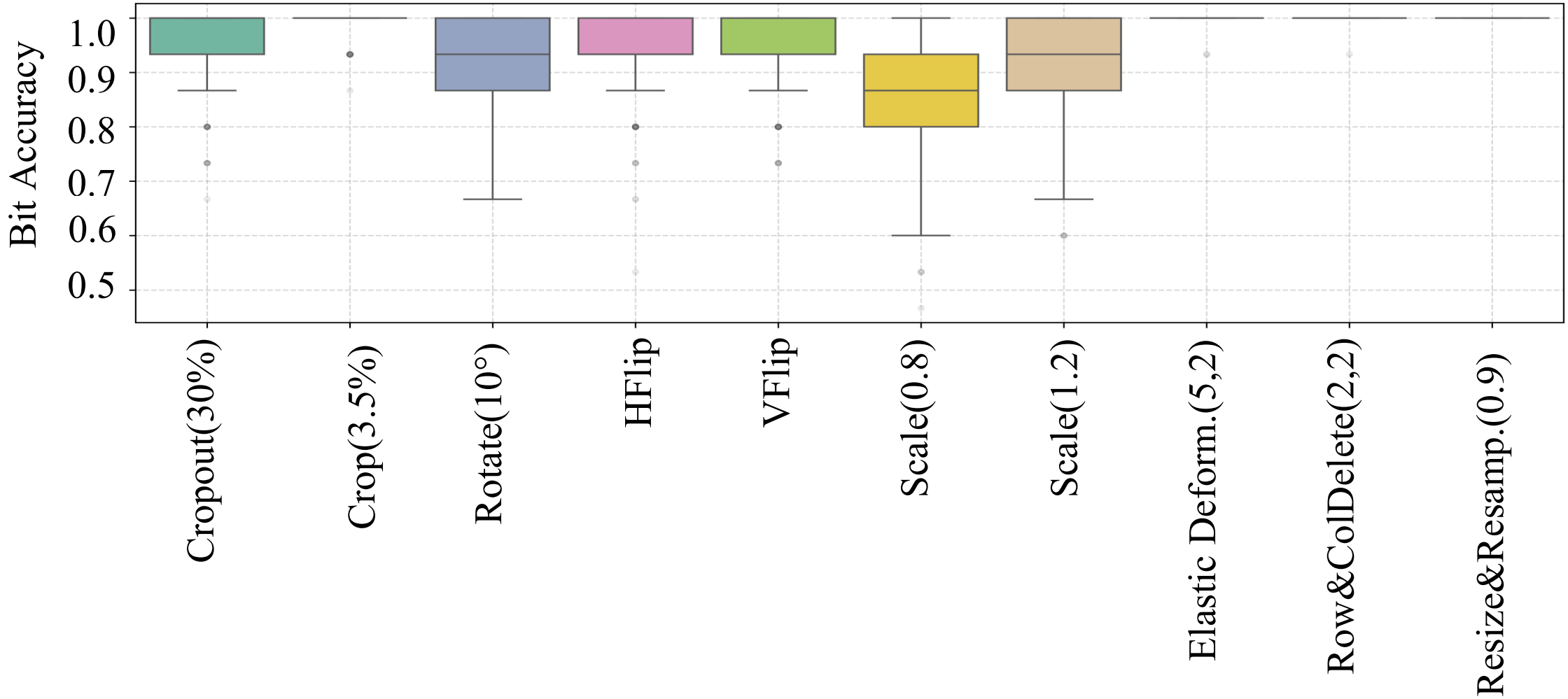


Figure 7: Bit accuracy under geometric perturbations

Qualitative Results

- Cover image, watermarked images, and re-watermarked images from the proposed model

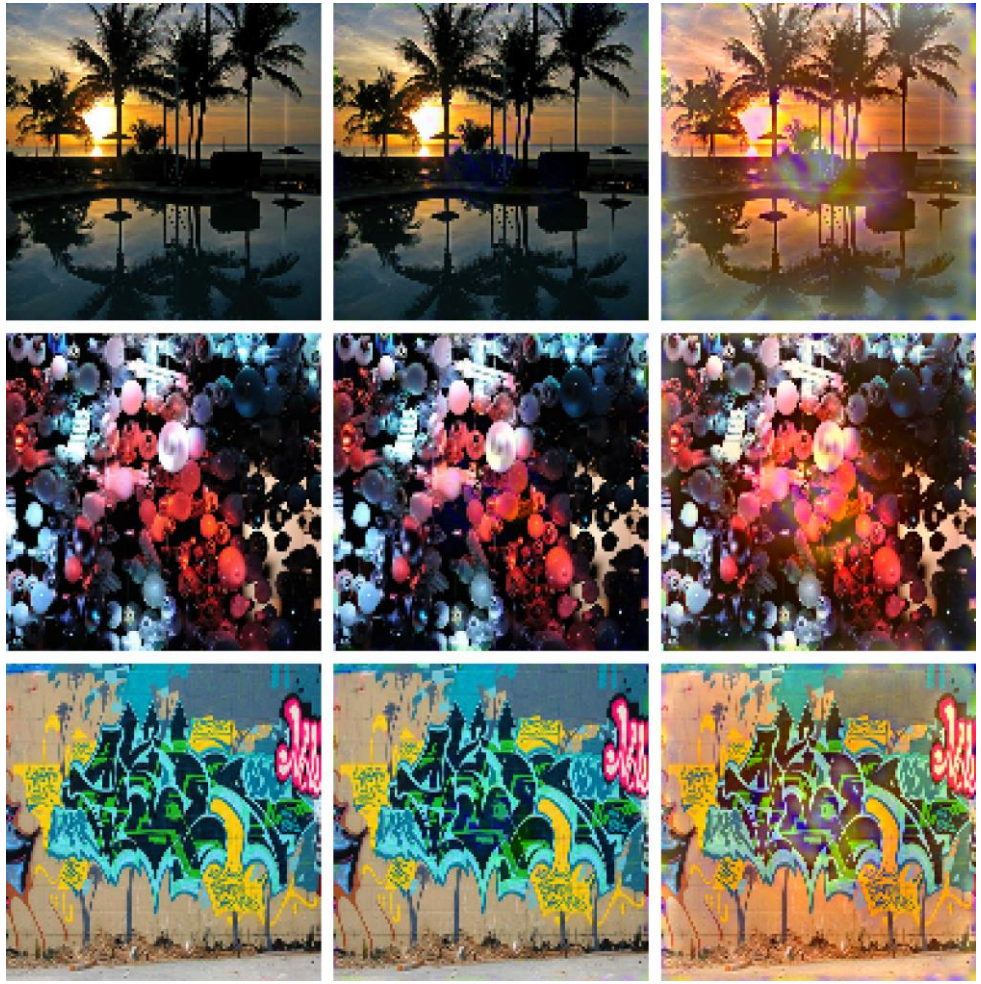


Figure 8: Visual Examples

Thank You!