arXiv:2608.15113v1 [cs.CV] 15 Aug 2026

Fast Test-Time Refinement for Robust Learned Image Compression

Jiaming Liang
University of Macau
Macau, China
chinaliangjm@gmail.com

Chi-Man Pun✉
University of Macau
Macau, China
cmpun@um.edu.mo

Weisi Lin
Nanyang Technological University
Singapore
wslin@ntu.edu.sg

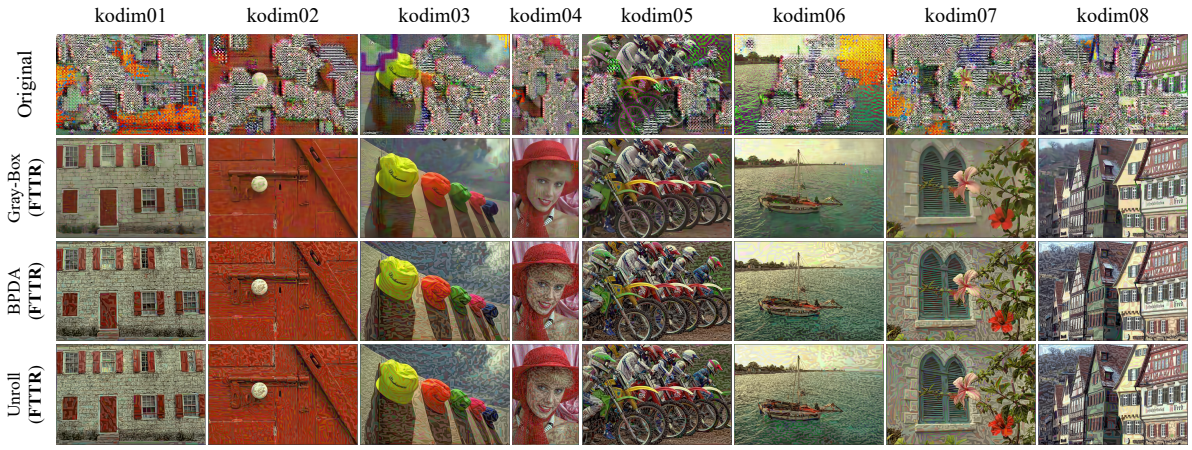

Figure 1: Reconstruction of examples generated by PGD (budget 16/255, 400 steps). *Original*: attack and reconstruction on the same undefended LIC system f. *Gray-Box*: attack on f and reconstruction by FTTR-equipped f. *BPDA* and *Unroll*: white-box adaptive attacks on FTTR-equipped f, with STE approximation and exact gradients, respectively. Example based on DCAE [40].

Abstract

Learned image compression (LIC) has demonstrated remarkable rate-distortion (RD) performance in benign settings. However, the high representational capacity endowed by deep neural networks (DNNs) comes at the expense of increased adversarial vulnerability. This hinders their adoption as trusted standardized codecs. Recent work has sketched test-time refinement (TTR) as a defense in gray-box scenarios, despite its original purpose of improving benign RD performance. Unfortunately, extensive iterations of TTR incur prohibitive overhead, while the robustness mechanism lacks theoretical understanding. Moreover, TTR has not been evaluated in white-box settings or against attacks beyond ℓ_2-bounded rate and untargeted distortion objectives. To bridge these gaps, we present a systematic study. Our study reveals an **Asymmetric Adversarial Trajectory (AAT)** property in LIC systems: transitioning from adversarial to benign regions is significantly easier than the reverse process, where adversarial examples can often be roughly recovered within only 1–2 steps. We provide a two-dimensional **Tube Model** to explain this phenomenon. Based on AAT, we propose a **Fast Test-Time Refinement (FTTR)** framework for practical and robust LIC systems. We establish that the robustness arises from the contraction of adversarial regions induced by the **Input-as-Label** property of LIC systems, rather than from obfuscated gradients. Extensive evaluations with diverse strong adaptive attacks across multiple LIC systems demonstrate the promise of the proposed FTTR framework. The code is available at https://github.com/chinaliangjiaming/FTTR.git.

CCS Concepts

• **Computing methodologies** → **Learning latent representations**; **Reconstruction**; **Image representations**; • **Security and privacy** → **Software and application security**.

Keywords

Machine Learning Security, Learned Image Compression, Adversarial Robustness, Test-Time Optimization

1 Introduction

Background. Learned image compression (LIC) [5, 6] incorporates deep neural networks (DNNs) to enhance representation capability, significantly reducing compression redundancy and achieving superior rate-distortion (RD) performance [28, 61]. However, the inherent high-dimensional vulnerability of DNNs [18, 54] is also inherited by LIC systems, making them more susceptible to adversarial threats than conventional hand-crafted codecs such as JPEG [57], JPEG 2000 [49], HEVC/H.265 [53], and VVC/H.266 [7].

Recent studies have demonstrated that imperceptible adversarial perturbations can induce rate collapse [33, 58], reconstruction distortion [29, 58], or even manipulated reconstruction [11, 37], as well as downstream degradation [52] in LIC systems. As a critical component of communication systems, compression requires strong reliability guarantees. In particular, standardized compression systems are expected to fully disclose all technical details,

which exposes LIC systems to adversaries [11]. Therefore, developing trustworthy LIC systems for secure deployment is imperative.

Despite this pressing need, research on robust LIC systems remains limited. One line of research improves the LIC robustness through adversarial training [9, 11, 33, 58, 63]. However, in white-box settings, attackers can circumvent such defenses by generating adversarial examples directly against the adversarially trained model [55]. Another direction [50, 58] keeps the model structure and parameters fixed, while mitigating adversarial inputs and optimizing latent representations at test time. Among them, static input transformations [50] provide limited protection against attacks under white-box settings, as attackers can generate tailored perturbations by incorporating the transformation into the attack pipeline. In contrast, test-time refinement (TTR) [14], as sketched in [58], provides dynamic protection against adversarial attacks through optimization, showing potential against white-box threats.

Specifically, TTR optimizes a purifier at test time via gradient descent based on the RD loss between the reconstructed image and the input image. The purifier directly or indirectly refines the latent and hyper-latent representations, and the optimized representations are then used for storage or transmission. TTR was originally proposed to improve benign RD performance at test time, particularly for domain adaptation scenarios, such as adapting codecs trained on natural images to screen content compression. [58] extends this idea by transferring the framework to adversarial inputs. Benefiting from its dynamic optimization property, TTR has the potential to make LIC systems robust against white-box attacks.

Gaps & Challenges. Unfortunately, [58] only sketches TTR as a defense benchmark in the experiments without systematic investigation, leaving several important issues unaddressed. **(I)** Conventional TTR requires a large number of optimization iterations, each involving backpropagation, making it computationally impractical. **(II)** The theoretical foundation of TTR as a defense remains unexplored, leaving its robustness mechanism unclear. **(III)** Existing evaluations of TTR as a defense have been confined to a narrow range of settings. **(III-a)** [58] considers only a gray-box setting, where TTR is disabled during attack generation and enabled only at test time. As a result, its robustness against white-box adaptive attacks remains unknown, casting doubt on whether the observed robustness stems from genuine effects or obfuscated gradients [4]. **(III-b)** TTR has primarily been evaluated under untargeted distortion and rate collapse attacks, with limited study on more recent threats such as semantic manipulation [37] and downstream task degradation [52]. **(III-c)** TTR has only been evaluated under ℓ_2 attacks, lacking a comprehensive assessment under other constraints.

Theory & Methodology. To bridge these gaps, we present a systematic study of TTR as an adversarial defense.

(I) Our study uncovers an intriguing property of LIC systems, termed Asymmetric Adversarial Trajectory (AAT) (Section 3.3). Specifically, we find that recovering adversarial examples of LIC systems often requires substantially fewer optimization steps than generating them. In most LIC systems, when the refinement strength is sufficient, the recovery process can be reduced to only 1–2 steps, as illustrated in Figure 2. We attribute the AAT property to the loss landscape shaped by the identity mapping property [37] of LIC

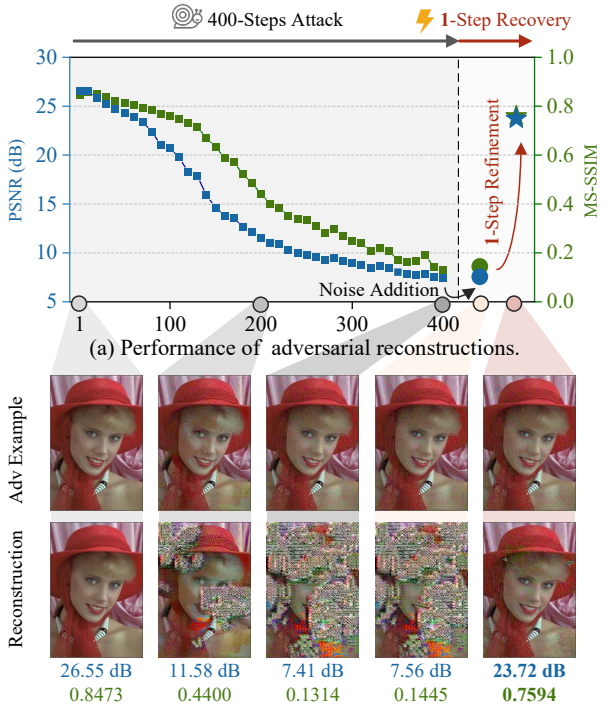

Figure 2: Our revealed Asymmetric Adversarial Trajectory (AAT) property of LIC systems: Generating effective adversarial examples against LIC systems requires numerous iterations, whereas their restoration to benign samples can typically be achieved within only 1–2 steps. The illustrated case is based on DCAE [40] and PGD attack [43].

systems, and further propose a two-dimensional Tube Model (Section 3.3) to explain this phenomenon. Building upon the AAT property, we propose Fast Test-Time Refinement (FTTR) (Section 3.4), an efficient framework of adversarial defense for LIC systems.

(II) We theoretically show that TTR provides genuine robustness gains rather than obfuscated gradients. Unlike other deep learning systems, LIC systems possess the Input-as-Label property (Section 3.5), whereby the adversarial example itself acts as the target label during optimization. This unique property enables TTR to minimize the loss for each input, thereby contracting adversarial regions. Based on the AAT property, our FTTR is an efficient variant of TTR that achieves a favorable robustness-efficiency trade-off.

(III) We evaluate the robustness of FTTR under a wide range of settings (Section 4). **(III-a)** Both white-box and gray-box settings are considered. In the white-box setting, we evaluate two strong adaptive attacker assumptions: Unroll [3] and BPDA [4]. **(III-b)** Our experiments cover diverse threats, including untargeted distortion, semantic manipulation, rate collapse, and downstream degradation. **(III-c)** Furthermore, we comprehensively evaluate the performance against both ℓ_2- and ℓ_∞-bounded attacks.

Results. Extensive experiments demonstrate that LIC systems equipped with our FTTR achieve strong white-box robustness with high efficiency. Specifically, under a 400-step ℓ_∞ white-box PGD

attack with a budget of 16/255, the average PSNR and bpp without FTTR are 9.90 dB and 17.97, respectively. In contrast, with 1-step FTTR, the average PSNR increases to **24.58**/**25.00** dB (**+14.68**/**15.10** dB) under BPDA/Unroll adversaries, while the average bpp decreases to **11.83**/**11.57** (**−6.14**/**6.40**), respectively. Under a 1000-step ℓ_2 white-box C&W attack with $c = 1000$, the average PSNR and bpp without FTTR are 7.59 dB and 15.19, respectively. In contrast, with 1-step FTTR, the average PSNR and bpp are **26.71** dB (**+19.12** dB) and **7.58** (**−7.61**) under the BPDA adversary. Furthermore, under a white-box PGD-based LIC-triggered downstream classification attack with perturbation of 8/255, our FTTR boosts the downstream classification accuracy from 0.4% to **28.1**% (**+27.7**%).

Our contributions can be summarized as follows:

- To the best of our knowledge, this is the first systematic study of TTR as an adversarial defense.
- We reveal the AAT property of LIC systems, showing that recovering adversarial examples to benign regions requires few steps in most LIC systems. We propose the Tube Model to illuminate this intriguing phenomenon.
- Building upon the AAT property, we propose FTTR, an efficient framework for leveraging TTR in adversarial defense.
- We theoretically show that the unique Input-as-Label property of LIC systems enables TTR to contract adversarial regions, thereby improving robustness. Our FTTR is an efficient variant of TTR that preserves strong robustness while substantially improving efficiency.
- Extensive experiments under diverse settings demonstrate that LIC systems equipped with our FTTR achieve strong robustness efficiently, highlighting its potential.

2 Related Work

2.1 Learned Image Compression

LIC follows the transform coding framework, which consists of transform, quantization, and entropy coding. LIC integrates DNNs into the analysis–synthesis transform [5] to improve representation capacity and reduce redundancy, and into entropy modeling to accurately estimate latent distributions [6]. Consequently, LIC achieves superior RD performance over conventional handcrafted compression codecs. Efforts to improve LIC models can be broadly grouped into three directions: optimization of the analysis–synthesis architecture [12, 16, 17, 26, 35, 38, 61, 64], quantization [1, 2, 20, 47, 62], and entropy modeling [22, 23, 25, 27, 30, 34, 36, 45, 48].

2.2 Adversarial Threats on LIC

Despite the impressive performance of LIC systems in benign settings, their reliance on DNNs makes them inherently susceptible to adversarial attacks. The adversarial threats to LIC systems can be divided as follows. (1) The first category causes rate collapse [11, 33, 39, 58, 60], resulting in a significant increase in storage and transmission overhead. (2) The second category targets reconstruction fidelity. Depending on whether the adversarial reconstruction is manipulated toward a specific outcome, it can be further divided into untargeted reconstruction distortion [11, 29, 33, 42, 51, 58, 60] and semantic manipulation [11, 37]. (3) The last is downstream degradation [52], where the adversary seeks to induce downstream task failures by manipulating reconstructions of LIC systems.

2.3 Adversarial Defenses on LIC

LIC systems are typically deployed in a white-box setting, where technical details are publicly disclosed and adversaries possess full knowledge of the system. Consequently, white-box robustness is of particular importance for LIC. Unfortunately, efforts to enhance LIC robustness remain scarce. Most existing work [9, 11, 33, 58, 63] focuses on adversarial training [18]. They alleviate vulnerability by incorporating adversarial examples during training. However, adversarial training struggles to eliminate adversarial vulnerability. Although it partially patches the vulnerabilities of the original LIC system, adversarial training may inadvertently introduce new vulnerabilities, causing the adversarial region to shift rather than contract. Attackers can then optimize against the adversarially trained model to craft adaptive examples that bypass the defense [55].

In contrast, a limited number of studies focus on mitigating adversarial effects through input transformations, thereby projecting adversarial examples back onto the benign data manifold. [50] places a static transformation before the LIC system and an inverse-transformation module after it. By comparing the reconstruction losses of multiple candidate transformation pairs, the method adopts the least-loss reconstruction as the final output. While effective in the black-box setting, the defense can be circumvented by adaptive attacks in the white-box setting, where all candidate transformation pairs are known to the adversary. Unlike this method, [58] pioneers the use of TTR based on the optimization for dynamic defense.

2.4 Test-Time Refinement

TTR [14] was originally proposed to improve in-distribution RD performance of LIC systems. Subsequently, TTR was extended to address the degradation of RD performance under distribution shifts, such as when LIC systems trained on natural images are applied to out-of-distribution data including illustrations or screen content images [10, 56]. With both the architecture and model parameters fixed, TTR performs iterative test-time optimization on the latent representation [10, 14, 19, 41, 56, 59] and side information [10, 19, 59], thereby obtaining a refined outcome.

However, TTR has long been confined to benign settings. [58] were the first to employ TTR as a defense. Unfortunately, TTR was only sketched as a gray-box defense benchmark in the experiments, leaving its potential for adversarial defense largely unexplored. Several critical questions remain unanswered, including the impracticality of multi-step optimization, the lack of a theoretical foundation for its robustness, and the absence of rigorous evaluations against diverse adaptive settings. In this work, we provide a systematic investigation of these issues and address them.

3 Theory and Methodology

3.1 Preliminaries

LIC Systems. For theoretical analysis, we abstract LIC systems into a general framework. An LIC system f is composed of an encoder E, a quantizer Q, and a decoder D:

$$f = D \circ Q \circ E, \tag{1}$$

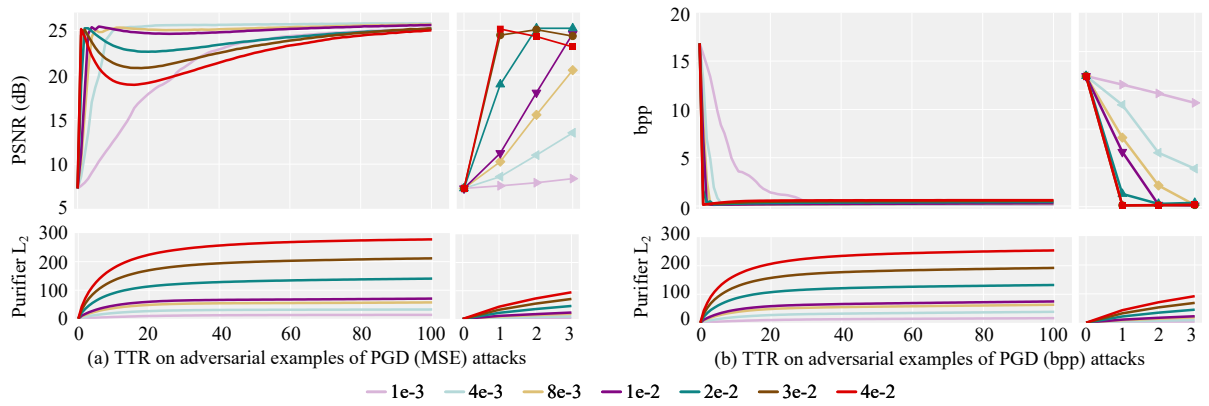

Figure 3: Performance of Adam-based TTR over iterations on PGD-generated MSE and bpp adversarial examples for DCAE. X-axis denotes the number of TTR iterations. The right subplot in each case provides a zoomed-in view of the first three TTR iterations. Colors indicate different Adam learning rates. *Purifier* L_2 denotes the ℓ_2 norm of the corresponding purifier.

where $\circ$ indicates composition. The encoder E encodes an input image $\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}$ into a latent representation $\boldsymbol{y}$ and a hyper-latent representation $\boldsymbol{z}$:

$$\left[\boldsymbol{y}, \boldsymbol{z}\right] = E(\boldsymbol{x}), \tag{2}$$

which are then quantized by decoder Q into $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$ for storage and transmission:

$$\left[\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}\right] = Q(\boldsymbol{y}, \boldsymbol{z}). \tag{3}$$

The decoder D reconstructs the quantized latent representation $\hat{\boldsymbol{y}}$ into the reconstructed image $\hat{\boldsymbol{x}}$:

$$\hat{\boldsymbol{x}} = D(\hat{\boldsymbol{y}}). \tag{4}$$

The overall objective $\mathcal{L}$ consists of the bitrate $\mathcal{R}$ and distortion $\mathcal{D}$, which are balanced via a Lagrange multiplier λ:

$$\mathcal{L} = \underbrace{r(\hat{\boldsymbol{y}}) + r(\hat{\boldsymbol{z}})}_{\mathcal{R}} + \lambda \cdot \underbrace{d(\boldsymbol{x}, \hat{\boldsymbol{x}})}_{\mathcal{D}}, \tag{5}$$

where r measures the bit rate, and d is a distortion measure.

Test-Time Refinement. To adapt to the test domain and further improve RD performance, TTR iteratively optimizes $\boldsymbol{y}$ and $\boldsymbol{z}$ via gradient descent on the RD objective at test time, yielding $\boldsymbol{y}^*$ and $\boldsymbol{z}^*$ as refined compression representations[1]:

$$\boldsymbol{y}^*, \boldsymbol{z}^* = \arg\min_{y,z} r(\boldsymbol{y}) + r(\boldsymbol{z}) + \lambda \cdot d(\boldsymbol{x}, \hat{\boldsymbol{x}}). \tag{6}$$

$\boldsymbol{y}^*$ and $\boldsymbol{z}^*$ are then quantized by Q for storage and transmission:

$$\left[\hat{\boldsymbol{y}}^*, \hat{\boldsymbol{z}}^*\right] = Q(\boldsymbol{y}^*, \boldsymbol{z}^*), \tag{7}$$

and decoded by D to yield the refined reconstruction $\hat{\boldsymbol{x}}^*$:

$$\hat{\boldsymbol{x}}^* = D(\hat{\boldsymbol{y}}^*). \tag{8}$$

In this formulation, we do not explicitly define the purifier. Instead, it is implicitly embodied in the optimization process of Equation 6. Specifically, the purifier can transform $\boldsymbol{x}$ into $\boldsymbol{x}^*$, thereby producing $\boldsymbol{y}^*$ and $\boldsymbol{z}^*$. Alternatively, it can directly optimize $\boldsymbol{y}$ and $\boldsymbol{z}$. Consequently, the purifier corresponds to $\boldsymbol{x}^* - \boldsymbol{x}$, $\boldsymbol{y}^* - \boldsymbol{y}$, and $\boldsymbol{z}^* - \boldsymbol{z}$.

[1]Note that r in Equation 5 denotes the actual bit rate measured from the quantized representations during evaluation, whereas r in Equation 6 represents the estimated bit rate used for optimization.

3.2 Threat Model

The prior study [58] presents TTR as a defense exclusively under a gray-box threat model, where TTR is enabled only at test time and excluded from the attack pipeline. As a result, adaptive attacks are not considered, leaving the robustness against fully informed adversaries unknown. This threat model is unrealistic and inconsistent with Kerckhoffs's principle [46], since the design of standardized compression codecs is publicly available. Therefore, besides the gray-box threat, we further consider the white-box threat.

Let f_d denote the LIC system f with an integrated defense. The adversary is assumed to have full knowledge of f_d and can thus launch adaptive attacks $A \in \mathcal{A}$. For adaptive adversaries, we consider two attack assumptions, BPDA [4] and Unroll [3]. Detailed descriptions are provided in the appendix. The perturbations

$$\boldsymbol{\delta}_a = A(f_d, \boldsymbol{x}) \tag{9}$$

are constrained to the input, and the adversarial examples

$$\boldsymbol{x}_{\text{adv}} = \boldsymbol{x} + \boldsymbol{\delta}_a \tag{10}$$

are processed by a trusted third party for compression, storage, and transmission. Man-in-the-middle (MITM) attacks are ruled out, preventing the adversary from tampering with the latent and hyper-latent representations. To ensure imperceptibility, perturbations $\boldsymbol{\delta}_a$ are constrained within an ℓ_p norm ball of ϵ_a. In this paper, we consider the two widely studied cases, namely, $p = 2$ and $p = \infty$. The adversarial objectives considered include compression rate $\mathcal{R}$, distortion $\mathcal{D}$, and downstream-task performance $\mathcal{P}$.

3.3 Asymmetric Adversarial Trajectory of LIC

Test-Time Refinement for Adversarial Defense. [58] first introduced TTR as a defense benchmark for experiments. Specifically, they apply an iteratively refined purifier $\boldsymbol{\delta}_d^*$ to adversarial examples $\boldsymbol{x}_{\text{adv}}$ for generating purified examples:

$$\boldsymbol{x}_{\text{pur}} = \boldsymbol{x}_{\text{adv}} + \boldsymbol{\delta}_d^*, \tag{11}$$

where

$$\boldsymbol{\delta}_d^* = \arg\min_{\delta_d} \mathcal{L}_{\text{pur}}, \tag{12}$$

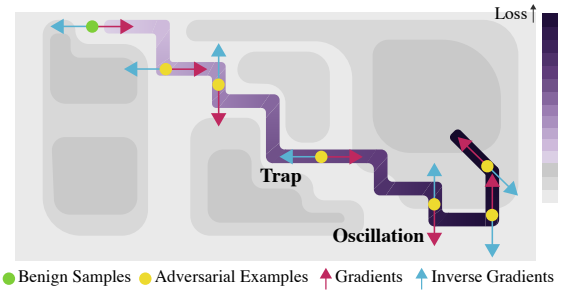

Figure 4: Hypothesized Tube Model for the two-dimensional structure of adversarial regions in LIC systems.

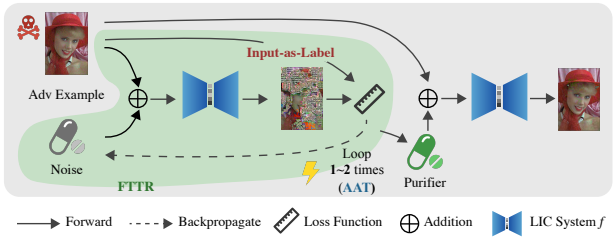

Figure 5: Pipeline of our FTTR for adversarial defense.

and

$$\mathcal{L}_{\text{pur}} = r\big(E(\boldsymbol{x}_{\text{adv}} + \boldsymbol{\delta}_d)\big) + \lambda \cdot d\big(\boldsymbol{x}_{\text{adv}}, f(\boldsymbol{x}_{\text{adv}} + \boldsymbol{\delta}_d)\big). \tag{13}$$

The purifier $\boldsymbol{\delta}_d^*$ is then propagated to indirectly optimize $\boldsymbol{y}$ and $\boldsymbol{z}$:

$$\boldsymbol{y}^*, \boldsymbol{z}^* = E(\boldsymbol{x}_{\text{pur}}). \tag{14}$$

It should be noted that latent and hyper-latent representations exhibit model-dependent scales, especially under adversarial settings, latent optimization may result in unstable updates. In contrast, input images are normalized, and input optimization provides stable parameterization. Therefore, we follow this strategy in this work.

Asymmetric Adversarial Trajectory. We begin our discussion with the gray-box threat model. To solve Equation 12, [58] performs gradient descent via Adam [31] with a small learning rate $\alpha = 0.01$ over multiple iterations. While numerous iterations are justified when optimizing benign inputs, they are unnecessary for the intent of defense. To illustrate this, we generate the MSE and bpp adversarial examples from `kodim04.png` in the Kodak [32] by 400 iterations of PGD with a budget of 16/255 and a step size of 2/255. TTR is then performed for 100 steps with Adam under different α. DCAE [40] is adopted as the tester. Results are shown in Figure 3.

From the figure, TTR with Adam under different learning rates α converges to similar performance within 100 steps, indicating that learning rates α have limited impact given sufficient iterations. However, under few-step optimization, learning rates α play a pivotal role for defense. Small α yields stable but slow convergence, whereas large α reaches near-convergent performance within 1–2 steps, albeit with transient instability followed by recovery. Although this observation is based on $\boldsymbol{x}_{\text{adv}}^{(400)}$, extensive experiments find it consistently holds for adversarial examples generated at arbitrary time steps t. Therefore, while generating adversarial examples in LIC systems is strenuous, restoring them to benign samples is remarkably effortless. We refer to this intriguing property of LIC systems as Asymmetric Adversarial Trajectory (AAT).

This naturally leads to the following question. (Q1) Why can large α reach near-convergent performance in even a single step? (or Why does AAT exist?) (Q2) Why does the performance subsequently drop sharply before recovering? The answer to (Q2) is relatively straightforward. Adam is designed for iterative convergence and inherently assumes a large number of iterations. Although its first update aligns with the gradient direction (see Equation 15), subsequent updates in the early iterations may deviate from the gradient. As a result, with a large α and low iterations, the optimizer may step in an incorrect direction, leading to degraded performance. Regarding (Q1), we next provide a detailed analysis.

Hypothesized Tube Model. When $\boldsymbol{\delta}_d^{(0)}$ is initialized as $\mathbf{0}$ and the stability constant $\epsilon \ll |\nabla_{\boldsymbol{\delta}_d} \mathcal{L}_{\text{pur}}|$, the first Adam update reduces to a sign gradient descent step with α (see Appendix for the deduction):

$$\boldsymbol{\delta}_d^{(1)} = -\alpha \cdot \operatorname{sgn}\big(\nabla_{\boldsymbol{\delta}_d^{(0)}} \mathcal{L}_{\text{pur}}\big). \tag{15}$$

In other words, at the first TTR iteration, Adam moves an ℓ_∞ distance of α along the negated gradient direction, i.e., away from the adversarial region. Combined with the revealed AAT property of LIC systems, this suggests that for adversarial examples $\boldsymbol{x}_{\text{adv}}^{(t)}$ generated at arbitrary attack steps t, they can escape the adversarial region and return to the benign region via a 1-step sign gradient descent with a sufficiently large α. This indicates that the adversarial region has extremely limited thickness along the negated gradient direction. Otherwise, a single-step update would be insufficient to escape the region if its thickness were large.

From a two-dimensional perspective, this analysis suggests that the adversarial region of LIC systems is likely to exhibit a thin, highly curved, and tube-like geometric structure. We refer to this hypothesized two-dimensional geometry of the adversarial region of LIC systems as Tube Model. As illustrated in Figure 4, due to the highly curved geometry of the adversarial region, linear updates of the adversarial perturbation $\boldsymbol{\delta}_a$ fail to reach effective adversarial examples within a few iterations, and instead require numerous small-step updates that progressively follow the loss-increasing trajectory. In contrast, for most points, one or two large-step updates along the negated gradient direction are sufficient to escape the adversarial region, which is consistent with the AAT property revealed above. Moreover, the Tube Model also accounts for the previously observed adversarial behaviors in LIC systems. For example, PGD^2-GSM [37] exhibits pronounced oscillations when attacking with large step sizes. This can be attributed to step sizes exceeding the local straight-segment length of the tube-like adversarial region, leading to overshooting into benign regions and consequently inducing oscillatory dynamics. In summary, Tube Model provides a new geometric perspective on the adversarial region of LIC systems.

3.4 Fast Test-Time Refinement for Defense

Therefore, numerous iterations are unnecessary when TTR is deployed for defense purposes. In practice, TTR for defense typically

Algorithm 1 Compression with Fast Test-Time Refinement (FTTR)

Input: Adversarial example $\boldsymbol{x}_{\text{adv}}$, LIC system f
Parameter: Radius s of purifier $\boldsymbol{\delta}_d$, iteration number K
Output: Optimized $\boldsymbol{y}^*$ and $\boldsymbol{z}^*$

Initialize $\boldsymbol{\delta}_d^{(0)} \sim \text{Uniform}(-s, s)$
for $k = 1$ to K **do**
  $\boldsymbol{g}^{(k)} = \text{sgn}\big(\nabla_{\boldsymbol{\delta}_d^{(k-1)}} \mathcal{L}_{\text{FTTR}}^{(k)}\big)$
  $\boldsymbol{\delta}_d^{(k)} = \text{Clip}_{(-s,s)}\big(\boldsymbol{\delta}_d^{(k-1)} - s \cdot K^{-\text{sgn}(k-1)} \cdot \boldsymbol{g}^{(k)}\big)$
end for
# The following conditional branch is only used in practice to
# ensure that benign performance is not degraded.
if $\mathcal{L}_{\text{FTTR}}(\boldsymbol{x}_{\text{adv}}, \boldsymbol{\delta}_d^{(K)}) < \mathcal{L}_{\text{FTTR}}(\boldsymbol{x}_{\text{adv}}, \boldsymbol{0})$ **then**
  return $E\big(\boldsymbol{x}_{\text{adv}} + \boldsymbol{\delta}_d^{(K)}\big)$
else
  return $E\big(\boldsymbol{x}_{\text{adv}}\big)$
end if

requires only a few iterations, or even a single-step update. Motivated by this insight, we redesign TTR into a lightweight defense framework, termed Fast Test-Time Refinement (FTTR).

FTTR initializes the purifier $\boldsymbol{\delta}_d^{(0)}$ with uniform noise rather than zeros. This initialization incurs negligible overhead while nudging adversarial examples away from the adversarial region. The purifier is then refined through one or a few optimization steps. The refinement loss follows Equation 13 to jointly optimize multiple objectives. Specifically, to defend against compression rate collapse:

$$\mathcal{L}_{\text{BPP}}^{(k)} = r\big(E\big(\boldsymbol{x}_{\text{adv}} + \boldsymbol{\delta}_d^{(k-1)}\big)\big), \tag{16}$$

For reconstruction quality, both the MSE loss:

$$\mathcal{L}_{\text{MSE}}^{(k)} = \text{MSE}\big(f\big(\boldsymbol{x}_{\text{adv}} + \boldsymbol{\delta}_d^{(k-1)}\big), \boldsymbol{x}_{\text{adv}}\big), \tag{17}$$

and the MS-SSIM loss:

$$\mathcal{L}_{\text{MS-SSIM}}^{(k)} = \text{MS-SSIM}\big(f\big(\boldsymbol{x}_{\text{adv}} + \boldsymbol{\delta}_d^{(k-1)}\big), \boldsymbol{x}_{\text{adv}}\big) \tag{18}$$

are adopted. When the refinement additionally aims to defend downstream attacks, an extra term:

$$\mathcal{L}_{\text{opt}}^{(k)} = J\big(\boldsymbol{x}_{\text{adv}}, \boldsymbol{\delta}_d^{(k-1)}\big) \tag{19}$$

is incorporated, where J is the criterion of the downstream task. Accordingly, the k^{th}-step iterative objective is summarized as:

$$\mathcal{L}_{\text{FTTR}}^{(k)} = \mathcal{L}_{\text{BPP}}^{(k)} + \lambda_1 \cdot \mathcal{L}_{\text{MSE}}^{(k)} - \lambda_2 \cdot \mathcal{L}_{\text{MS-SSIM}}^{(k)} + \lambda_3 \cdot \mathcal{L}_{\text{opt}}^{(k)}. \tag{20}$$

To prevent excessive distortion, the purifier is constrained within an ℓ_∞ ball of radius s. The first optimization step performs gradient descent along the negated sign gradient direction with step size s. Owing to the AAT property of LIC systems, this single large-step update is sufficient in most cases. When iteration $K > 1$, subsequent steps continue with a reduced step size of s/K. The complete procedure is summarized in Algorithm 1, while the overall pipeline is illustrated in Figure 5.

3.5 White-Box Robustness Mechanism

The preceding analysis focuses on the gray-box setting, where refinement is disabled during attacks but enabled at test time. As discussed in Section 3.2, white-box robustness is crucial for LIC systems. Therefore, we further examine the white-box robustness.

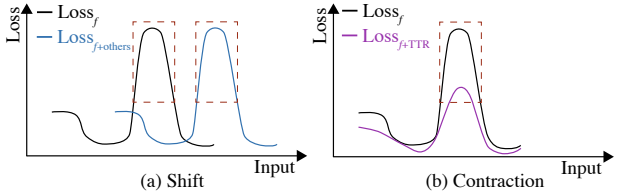

Figure 6: Illustrations on the shift and contraction of adversarial regions. TTR leverages the Input-as-Label to contract the adversarial region, enhancing white-box robustness.

For most deep learning tasks, achieving white-box robustness remains challenging, because the groundtruth of adversarial examples are inaccessible. For instance, in classification systems, an image of dog may be perturbed to be misclassified as kangaroo. The defender has no access to its true label dog, making recovery to a benign state intractable. Consequently, existing defenses rely on transformations, adversarial training, or related techniques to alleviate adversarial effects. Unfortunately, while they mitigate the original adversarial vulnerabilities, they often introduce new ones. In other words, they merely induce a shift of the adversarial region, rather than shrinking or eliminating it, which lies at the heart of the difficulty in achieving white-box robustness.

In contrast, LIC systems differ fundamentally from them. Under the control of a trusted third party over compression, storage, and transmission, the ground truth for any adversarial example is inherently available, since the reconstruction target is the adversarial example itself. We refer to this as the Input-as-Label property of LIC systems. TTR can serve as a white-box defense precisely because it exploits the Input-as-Label property of LIC systems. With an unlimited number of iterations, it is easy to see that TTR ensures that the loss can be reduced for any adversarial example, or at least remains non-increasing (because the trivial solution can be $\boldsymbol{\delta}_d^* = 0$):

$$\forall \boldsymbol{x}_{\text{adv}}, \quad \mathcal{L}\big(\boldsymbol{x}_{\text{adv}} + \boldsymbol{\delta}_d^*\big) \le \mathcal{L}\big(\boldsymbol{x}_{\text{adv}}\big). \tag{21}$$

Equation 21 implies that TTR can be viewed as a mapping acting on the LIC system f that never increases the loss. As illustrated in Figure 6(b), each sample either preserves its loss or moves to a lower-loss state. Therefore, the adversarial region is contracted rather than shifted. This indicates that TTR improves the intrinsic robustness of the LIC system, rather than creating an illusion.

Our FTTR is an efficient approximation of TTR. Benefiting from the revealed AAT property, extensive empirical results show that Equation 21 holds for most adversarial examples. However, since FTTR typically adopts a relatively large strength s, it may introduce excessive perturbations. When the adversarial perturbation is small, the negative effects introduced by FTTR may outweigh its benefits. Moreover, for similar reasons, FTTR may degrade benign performance. Therefore, we introduce a conditional branch to determine whether the purifier generated by FTTR provides a gain. If not, we set $\boldsymbol{\delta}_d^{(K)} = 0$, as shown in Algorithm 1. This ensures

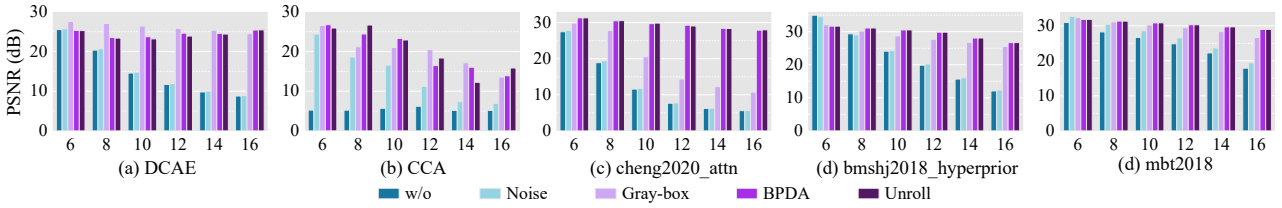

Figure 7: PSNR (dB) of PGD-attacked reconstructions under MSE. X-axis indicates perturbation budgets. *w/o*: attack/inference on undefended f. *Noise*: attack on undefended f, inference with noisy input. *Gray-box*: attack on undefended f, inference with FTTR. *BPDA/Unroll*: attack/inference with FTTR (see Appendix for details). The same notation applies hereafter.

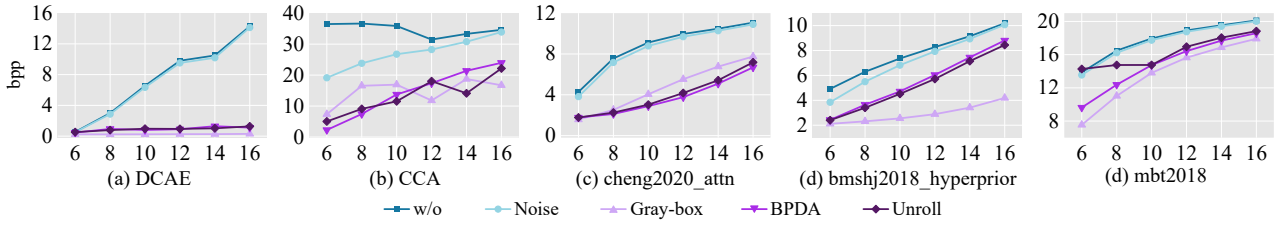

Figure 8: bpp of PGD-attacked reconstructions under bpp.

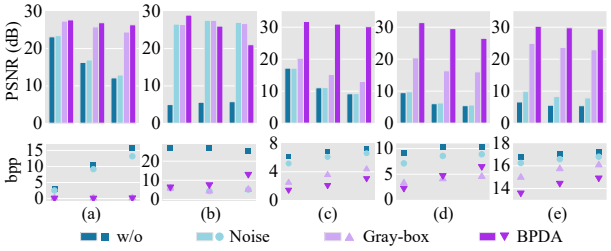

Figure 9: PSNR and bpp of C&W-attacked reconstructions under joint MSE and bpp. Order of (a)-(e) follows Figure 7.

that Equation 21 also holds for FTTR. To clearly present the intrinsic robustness of FTTR, we disable this conditional branch in our experiments and evaluate the iteratively optimized purifier.

In summary, this section theoretically established the white-box robustness mechanisms of TTR and our FTTR. The following extensive experiments will empirically validate them.

4 Experiments and Results

4.1 Setup

LIC systems. We evaluate our FTTR across multiple representative LIC systems. Specifically, for untargeted distortion attacks, bitrate collapse attacks, and downstream degradation attacks, we evaluate on two advanced LIC systems DCAE [40] and CCA [21], and three classical models cheng2020-Attn [13], bmshj2018-hyperprior [6], and mbt2018 [44]. For global semantic manipulation, following [37], we evaluate on DCAE, CCA, and HiFi-VRIC [8].

Datasets. Experiments are conducted on Kodak dataset [32], a standard benchmark for compression, which contains 24 RGB images of 512×768. For the downstream task of classification, we use the NIPS 2017 Adversarial Competition dataset [2].

Comprehensive Adversary Benchmarks. Prior study [58] evaluates only under the gray-box setting. In contrast, we establish a suite of adaptive adversary benchmarks to evaluate the white-box robustness, including BPDA [4] and Unroll [3]. BPDA performs standard forward propagation through FTTR but approximates FTTR as an identity mapping during backpropagation, enabling efficient perturbation generation with approximate gradients. In contrast, Unroll explicitly unfolds the optimization process to obtain exact gradients, but incurs substantial computational overhead due to second-order gradients. More details are provided in the appendix. Besides, we also evaluate FTTR under the gray-box setting.

Implementations. We set $\lambda_1 = 10^3$, λ_2=50, s=0.04. Except for downstream defenses, λ_3 is 0. We adopt 1-step FTTR with K=1.

Metrics. For reconstruction quality, we use pixel-level PSNR (dB), structural-level MS-SSIM, and semantic-level LPIPS and CLIP. For compression rate, we use bpp. Note that adversarial examples may cause excessively large bpp, making exact computation infeasible within a reasonable time. Therefore, we report the estimated bpp. For classification tasks, we use top-1 accuracy (%).

4.2 To ℓ_∞ Untargeted Distortion Attacks

This experiment evaluates the robustness of FTTR against ℓ_∞ untargeted distortion attacks based on PGD, where the attack objective is the MSE loss. We compare five settings: (1) *w/o*: both attack and inference are performed on the undefended LIC system f. (2) *Noise*: the attack is performed on f, while uniform noise with strength 0.04 is added before inference on f. (3) *Gray-Box*: the attack is performed on f, while inference is performed on the FTTR-equipped LIC system f_{FTTR}. (4) *BPDA*: adaptive white-box attacks are generated against f_{FTTR} using BPDA, followed by inference on f_{FTTR}. (5)

[2] https://github.com/anlthms/nips-2017.git

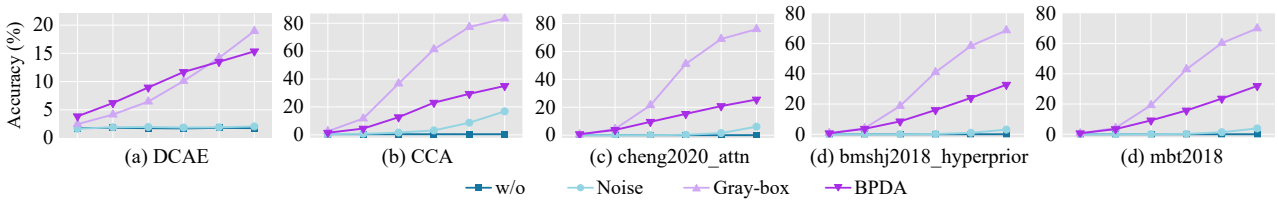

Figure 10: Classification accuracy (%) of LIC-triggered downstream attacks.

Table 1: Reconstruction performance on adversarial examples for high-resolution global semantic manipulation. *FTTR* denotes the *Gray-Box*. The best values are shown in bold.

		PSNR ↑	MS-SSIM ↑	LPIPS ↓	CLIP ↑	bpp ↓
DCAE	w/o	12.970	0.403	0.687	0.724	1.312
	Noise	19.146	0.593	0.658	0.830	0.291
	FTTR	**21.698**	**0.717**	**0.615**	**0.871**	**0.141**
CCA	w/o	15.704	0.579	0.380	0.786	4.084
	Noise	21.536	0.834	0.180	0.940	3.746
	FTTR	**24.415**	**0.912**	**0.076**	**0.980**	**1.970**
HiFi-VRIC	w/o	12.635	0.390	0.687	0.626	11.148
	Noise	16.464	0.618	0.530	0.748	9.718
	FTTR	**24.002**	**0.901**	**0.092**	**0.982**	**6.240**

Unroll: adaptive white-box attacks are generated against f_{FTTR} using Unroll, followed by inference on f_{FTTR}. The same protocol is used hereafter and is omitted for brevity. We generate adversarial examples with iterations $T = 400$, budgets $\epsilon \in \{6, 8, 10, 12, 14, 16\}/255$, and step size ϵ/T. The PSNR results are reported in Figure 7, with MS-SSIM, LPIPS, and CLIP in the Figure 12. The visualizations are in the Figure 13. At $\epsilon = 16/255$, the *w/o* setting yields an average PSNR of only 9.90 dB, while the FTTR-equipped models maintain 24.58(+14.68) dB and 25.00(+15.10) dB under adaptive white-box BPDA and Unroll, respectively. These results demonstrate strong white- and gray-box robustness of FTTR against ℓ_∞ MSE attacks.

4.3 To ℓ_∞ Compression Rate Attacks

This experiment evaluates the robustness of FTTR against ℓ_∞ compression rate attacks based on PGD, where the attack objective is the bpp loss. Experimental settings follow Section 4.2. Results in Figure 8 demonstrate that FTTR achieves large robustness gains against ℓ_∞ compression rate attacks. On average, at $\epsilon = 16/255$, FTTR reduces the bpp from 17.97 in the w/o setting to 11.83(−6.14) under BPDA and 11.57(−6.40) under Unroll. The robustness gains vary across LIC systems. Remarkably, for DCAE, the bpp is reduced from 14.29 to 1.102 under BPDA and 1.27 under Unroll. This indicates that certain LIC models can achieve more substantial robustness improvements when combined with FTTR, highlighting the potential of FTTR for trustworthy standardized LIC systems.

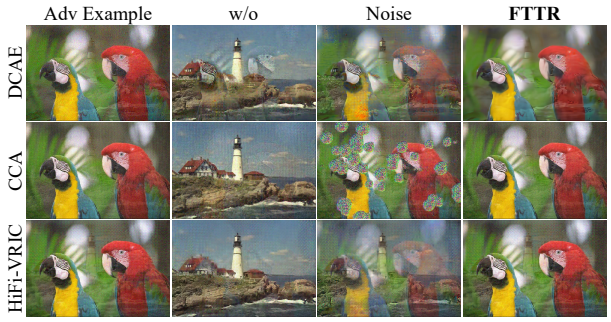

Figure 11: Adversarial examples and reconstructions under global semantic manipulation. *FTTR* denotes the *Gray-Box*.

4.4 To ℓ_2 Joint Rate-Distortion Attacks

We further evaluate FTTR against ℓ_2 joint R-D attacks based on C&W attack, where the objective is:

$$\max -\|\boldsymbol{\delta}_a\|_2 + c \cdot [c_1 \cdot r(E(\boldsymbol{x}_{\text{adv}})) + c_2 \cdot \text{MSE}(f(\boldsymbol{x}_{\text{adv}}), \boldsymbol{x})]. \quad (22)$$

c_1 and c_2 are set to 1 and 500, respectively. Unlike the standard C&W attack, we fix three representative c of 500, 750, 1000. The attack uses Adam with a learning rate of 10^{-3} for 1000 iterations. Results in Figure 9 and visualizations in Figure 14 demonstrate the robustness against ℓ_2 attacks. At $c = 1000$, on average, PSNR improves from 7.59 dB in the w/o setting to 26.71(+19.12) dB under BPDA, while bpp decreases from 15.19 to 7.58(−7.61).

4.5 To Global Semantic Manipulations

Beyond conventional attacks, we evaluate FTTR against the emerging global semantic manipulation (GSM). We use PGD^2-GSM [37] with $\epsilon = 0.10$, while keeping other settings the same as the original paper. Since GSM typically requires tens of thousands of iterations, we only consider the gray-box threat setting here. Quantitative results in Table 1 and visualizations in Figure 11 demonstrate that FTTR effectively resists this challenging attack.

4.6 To LIC-Triggered Downstream Attacks

We further evaluate FTTR against LIC-triggered downstream classification attacks, which generate adversarial examples that remain benign before LIC processing but cause downstream failures after compression and decompression. The attack is PGD with budget 8/255, step size 1/255 and iterations 40. The classification models include ResNet-50 [24] and ViT-B/16 [15]. The FTTR strength s is varied from 0.01 to 0.06 with an interval of 0.01. The classification loss is the cross-entropy loss, with λ_3 set to 1. Results in Figure 10

demonstrate that FTTR substantially improves robustness against LIC-triggered downstream attacks. At $s = 0.06$, our FTTR boosts the average downstream classification accuracy from 0.4% under *w/o* setting to 28.1%(+27.7%) under *BPDA* setting.

5 Conclusions and Limitations

Existing robustness studies on LIC focus on gray-box setting, while the white-box robustness of LIC systems remains unexplored. This work presents the first systematic study on the potential of TTR as a white-box defense. We reveal the Asymmetric Adversarial Trajectory (AAT) property of LIC systems that TTR does not require numerous iterations when used for defense purposes. For most LIC systems, a few steps are sufficient. We explain this phenomenon through the hypothesized Tube Model and propose the Fast Test-Time Refinement (FTTR) framework for practical defense deployment. Our analysis shows that TTR and FTTR exploit the Input-as-Label property of LIC systems to contract the adversarial region rather than shift it, leading to genuine robustness improvements. Even with only 1-step FTTR, our experiments demonstrate remarkable white-box robustness gains under various threats. Further optimization of FTTR, such as adaptive strength control, remains an interesting and promising direction for future research.

References

[1] Eirikur Agustsson and Lucas Theis. 2020. Universally quantized neural compression. *Advances in neural information processing systems* 33 (2020), 12367–12376.

[2] David Alexandre, Chih-Peng Chang, Wen-Hsiao Peng, and Hsueh-Ming Hang. 2019. Learned image compression with soft bit-based rate-distortion optimization. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1715–1719.

[3] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. 2016. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems* 29 (2016).

[4] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*. PMLR, 274–283.

[5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. 2017. End-to-end Optimized Image Compression. In *International Conference on Learning Representations*.

[6] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. 2018. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*.

[7] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. 2021. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 10 (2021), 3736–3764.

[8] Shilv Cai, Zhijun Zhang, Liqun Chen, Luxin Yan, Sheng Zhong, and Xu Zou. 2022. High-Fidelity Variable-Rate Image Compression via Invertible Activation Transformation. In *Proceedings of the 30th ACM International Conference on Multimedia*. ACM, 2021–2031.

[9] Zhi Cao, Youneng Bao, Fanyang Meng, Chao Li, Wen Tan, Genhong Wang, and Yongsheng Liang. 2024. Enhancing adversarial training with prior knowledge distillation for robust image compression. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3430–3434.

[10] Kecheng Chen, Pingping Zhang, Tiexin Qin, Shiqi Wang, Hong Yan, and Haoliang Li. 2025. Test-time adaptation for image compression with distribution regularization. In *International Conference on Learning Representations*, Vol. 2025. 95215–95232.

[11] Tong Chen and Zhan Ma. 2023. Toward robust neural image compression: Adversarial attack and model finetuning. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 12 (2023), 7842–7856.

[12] Yunuo Chen, Bing He, Zezheng Lyu, Hongwei Hu, Qunshan Gu, Yuan Tian, and Guo Lu. 2026. Adaptive Learned Image Compression with Graph Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12150–12161.

[13] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. 2020. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7939–7948.

[14] JCMSA Djelouah and Christopher Schroers. 2019. Content adaptive optimization for neural image compression. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, Vol. 2. 1–5.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[16] Donghui Feng, Zhengxue Cheng, Shen Wang, Ronghua Wu, Hongwei Hu, Guo Lu, and Li Song. 2025. Linear attention modeling for learned image compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 7623–7632.

[17] Runsen Feng, Zongyu Guo, Weiping Li, and Zhibo Chen. 2023. Nvtc: Nonlinear vector transform coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6101–6110.

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

[19] Tiansheng Guo, Jing Wang, Ze Cui, Yihui Feng, Yunying Ge, and Bo Bai. 2020. Variable rate image compression with content adaptive optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 122–123.

[20] Zongyu Guo, Zhizheng Zhang, Runsen Feng, and Zhibo Chen. 2021. Soft then hard: Rethinking the quantization in neural image compression. In *International Conference on Machine Learning*. PMLR, 3920–3929.

[21] Minghao Han, Shiyin Jiang, Shengxi Li, Xin Deng, Mai Xu, Ce Zhu, and Shuhang Gu. 2024. Causal context adjustment loss for learned image compression. *Advances in Neural Information Processing Systems* 37 (2024), 133231–133253.

[22] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. 2022. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5718–5727.

[23] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. 2021. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14771–14780.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[25] Qian He, Yuqi Li, Yiheng Jiang, Haotian Zhang, Li Li, and Dong Liu. 2026. Practical Image Compression with Energy-Guided Asymmetric Entropy Modeling. In *2026 Data Compression Conference (DCC)*. IEEE, 292–301.

[26] Wei Jiang, Peirong Ning, Jiayu Yang, Yongqi Zhai, Feng Gao, and Ronggang Wang. 2024. Llic: Large receptive field transform coding with adaptive weights for learned image compression. *IEEE Transactions on Multimedia* 26 (2024), 10937–10951.

[27] Wei Jiang, Jiayu Yang, Yongqi Zhai, Feng Gao, and Ronggang Wang. 2025. MLIC++: Linear complexity multi-reference entropy modeling for learned image compression. *ACM Transactions on Multimedia Computing, Communications and Applications* 21, 5 (2025), 1–25.

[28] Wei Jiang, Yongqi Zhai, Jiayu Yang, Feng Gao, and Ronggang Wang. 2026. Mlicv2: Enhanced multi-reference entropy modeling for learned image compression. *ACM Transactions on Multimedia Computing, Communications and Applications* 22, 4 (2026), 1–23.

[29] Nikolay I Kalmykov, Razan Dibo, Kaiyu Shen, Xu Zhonghan, Anh-Huy Phan, Yipeng Liu, and Ivan Oseledets. 2026. T-MLA: A targeted multiscale log–exponential attack framework for neural image compression. *Information Sciences* (2026), 123143.

[30] Jun-Hyuk Kim, Seungeon Kim, Won-Hee Lee, and Dokwan Oh. 2024. Diversify, contextualize, and adapt: Efficient entropy modeling for neural image codec. *Advances in Neural Information Processing Systems* 37 (2024), 45956–45974.

[31] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

[32] Eastman Kodak. 1993. Kodak Lossless True Color Image Suite (PhotoCD PCD0992). http://r0k.us/graphics/kodak/

[33] Jun Kurihara and Heming Sun. 2025. Efficient Adversarial Attack and Training on Learned Image Compression. In *2025 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2453–2458.

[34] Chao Li, Shanzhi Yin, Chuanmin Jia, Fanyang Meng, Yonghong Tian, and Yongsheng Liang. 2024. Multirate progressive entropy model for learned image compression. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 8 (2024), 7725–7741.

[35] Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. 2024. Frequency-aware transformer for learned image compression. In *International Conference on Learning Representations*.

[36] Yuqi Li, Haotian Zhang, Li Li, and Dong Liu. 2025. Learned image compression with hierarchical progressive context modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 18834–18843.

[37] Jiaming Liang, Chi-Man Pun, Weisi Lin, and Greta Seng Peng Mok. 2026. Control Your View: High-Resolution Global Semantic Manipulation in Learned Image Compression. *arXiv preprint arXiv:2605.08727* (2026).

[38] Jinming Liu, Heming Sun, and Jiro Katto. 2023. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14388–14397.

[39] Kang Liu, Di Wu, Yangyu Wu, Yiru Wang, Dan Feng, Benjamin Tan, and Siddharth Garg. 2023. Manipulation attacks on learned image compression. *IEEE Transactions on Artificial Intelligence* 5, 6 (2023), 3083–3097.

[40] Jingbo Lu, Leheng Zhang, Xingyu Zhou, Mu Li, Wen Li, and Shuhang Gu. 2025. Learned image compression with dictionary-based entropy model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 12850–12859.

[41] Yue Lv, Jinxi Xiang, Jun Zhang, Wenming Yang, Xiao Han, and Wei Yang. 2023. Dynamic low-rank instance adaptation for universal neural image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*. 632–642.

[42] Jingui Ma and Ronggang Wang. 2024. An imperceptible adversarial attack against reconstruction for learned image compression. In *2024 Data Compression Conference (DCC)*. IEEE, 573–573.

[43] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

[44] David Minnen, Johannes Ballé, and George D Toderici. 2018. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems* 31 (2018).

[45] David Minnen and Saurabh Singh. 2020. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3339–3343.

[46] Fabien AP Petitcolas. 2025. Kerckhoffs' principle. In *Encyclopedia of Cryptography, Security and Privacy*. Springer, 1342–1343.

[47] Alberto Presta, Enzo Tartaglione, Attilio Fiandrotti, and Marco Grangetto. 2025. Stanh: Parametric quantization for variable rate learned image compression. *IEEE Transactions on Image Processing* 34 (2025), 639–651.

[48] Yichen Qian, Ming Lin, Xiuyu Sun, Zhiyu Tan, and Rong Jin. 2022. Entroformer: A transformer-based entropy model for learned image compression. In *International Conference on Learning Representations*.

[49] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. 2002. The JPEG 2000 still image compression standard. *IEEE Signal processing magazine* 18, 5 (2002), 36–58.

[50] Myungseo Song, Jinyoung Choi, and Bohyung Han. 2024. A training-free defense framework for robust learned image compression. *arXiv preprint arXiv:2401.11902* (2024).

[51] Yang Sui, Zhuohang Li, Ding Ding, Xiang Pan, Xiaozhong Xu, Shan Liu, and Zhenzhong Chen. 2024. Reconstruction Distortion of Learned Image Compression with Imperceptible Perturbations. In *2024 Data Compression Conference (DCC)*. IEEE, 583–583.

[52] Yang Sui, Zhuohang Li, Ding Ding, Xiang Pan, Xiaozhong Xu, Shan Liu, and Zhenzhong Chen. 2024. Transferable Learned Image Compression-Resistant Adversarial Perturbations. In *2024 Data Compression Conference (DCC)*. IEEE, 582–582.

[53] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology* 22, 12 (2012), 1649–1668.

[54] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[55] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems* 33 (2020), 1633–1645.

[56] Koki Tsubota, Hiroaki Akutsu, and Kiyoharu Aizawa. 2023. Universal deep image compression via content-adaptive optimization with adapters. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2529–2538.

[57] Gregory K Wallace. 1991. The JPEG still picture compression standard. *Commun. ACM* 34, 4 (1991), 30–44.

[58] Chenhao Wu, Qingbo Wu, Haoran Wei, Lei Wang, Fanman Meng, King Ngi Ngan, Li Zhuo, and Hongliang Li. 2025. On the Adversarial Robustness of Learning-based Image Compression Against Rate-Distortion Attacks. *IEEE Transactions on Multimedia* (2025).

[59] Yibo Yang, Robert Bamler, and Stephan Mandt. 2020. Improving inference for neural image compression. *Advances in Neural Information Processing Systems* 33 (2020), 573–584.

[60] Yi Yu, Yufei Wang, Wenhan Yang, Shijian Lu, Yap-Peng Tan, and Alex C Kot. 2023. Backdoor attacks against deep image compression via adaptive frequency trigger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12250–12259.

[61] Yichi Zhang, Yuning Huang, and Fengqing Zhu. 2026. QARV++: An Improved Hierarchical VAE for Learned Image Compression. *IEEE Transactions on Circuits and Systems for Video Technology* (2026).

[62] Jing Zhou, Akira Nakagawa, Keizo Kato, Sihan Wen, Kimihiko Kazui, and Zhiming Tan. 2020. Variable rate image compression method with dead-zone quantizer. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition workshops*. 162–163.

[63] Tianyu Zhu, Heming Sun, Xiankui Xiong, Xuanpeng Zhu, Yong Gong, Yibo Fan, et al. 2024. Attack and defense analysis of learned image compression. *arXiv preprint arXiv:2401.10345* (2024).

[64] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. 2022. The devil is in the details: Window-based attention for image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 17492–17501.

A Proofs

A.1 Deduction of Equation 15

We follow the notations of the original Adam [31] paper and do not repeat the definitions here. In particular, the variable being optimized is the purifier $\boldsymbol{\delta}_d$ rather than the model parameters θ. The parameters are initialized as in the original Adam paper:

$$\boldsymbol{m}_0 = \mathbf{0},\ \boldsymbol{v}_0 = \mathbf{0}. \tag{23}$$

The gradient at time step k is:

$$\boldsymbol{g}_k = \nabla_{\boldsymbol{\delta}_d^{(k-1)}} \mathcal{L}_{\text{pur}}. \tag{24}$$

At time step k, Adam updates the purifier $\boldsymbol{\delta}_d^{(k)}$ as follows:

$$\boldsymbol{m}_k = \beta_1 \cdot \boldsymbol{m}_{k-1} + (1-\beta_1)\cdot \boldsymbol{g}_k,\ \hat{\boldsymbol{m}}_k = \frac{\boldsymbol{m}_k}{1-\beta_1^k}, \tag{25}$$

$$\boldsymbol{v}_k = \beta_2 \cdot \boldsymbol{v}_{k-1} + (1-\beta_2)\cdot \boldsymbol{g}_k^2,\ \hat{\boldsymbol{v}}_k = \frac{\boldsymbol{v}_k}{1-\beta_2^k}, \tag{26}$$

$$\boldsymbol{\delta}_d^{(k)} = \boldsymbol{\delta}_d^{(k-1)} - \alpha \cdot \frac{\hat{\boldsymbol{m}}_k}{\sqrt{\hat{\boldsymbol{v}}_k} + \epsilon}. \tag{27}$$

By setting $k = 1$, we obtain:

$$\boldsymbol{m}_1 = (1-\beta_1)\cdot \boldsymbol{g}_1,\ \hat{\boldsymbol{m}}_1 = \boldsymbol{g}_1, \tag{28}$$

$$\boldsymbol{v}_1 = (1-\beta_2)\cdot \boldsymbol{g}_1^2,\ \hat{\boldsymbol{v}}_1 = \boldsymbol{g}_1^2, \tag{29}$$

$$\boldsymbol{\delta}_d^{(1)} = \boldsymbol{\delta}_d^{(0)} - \alpha \cdot \frac{\boldsymbol{g}_1}{|\boldsymbol{g}_1| + \epsilon}. \tag{30}$$

Given $\boldsymbol{\delta}_d^{(0)} = \mathbf{0}$ and $\epsilon \ll |\nabla_{\delta_d} \mathcal{L}_{\text{pur}}|$, it follows that:

$$\boldsymbol{\delta}_d^{(1)} = -\alpha \cdot \operatorname{sgn}(\boldsymbol{g}_1). \tag{31}$$

B Supplementary Experiments

B.1 Details of Adversary Benchmarks

Unroll. In white-box scenarios, FTTR can be viewed as a standalone defense module preceding LIC system f:

$$f_d = f \circ \text{FTTR}. \tag{32}$$

For a sample $\boldsymbol{x}$ and t-moment perturbation $\boldsymbol{\delta}_a^{(t)}$, the attacker performs gradient ascent to optimize it:

$$\boldsymbol{\delta}_a^{(t+1)} = \boldsymbol{\delta}_a^{(t)} + \alpha \cdot \nabla_{\boldsymbol{\delta}_a^{(t)}} \mathcal{L}(f_d(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)}), \boldsymbol{x}). \tag{33}$$

We expand the gradient in Equation 33 as follows:

$$\begin{aligned}
&\nabla_{\boldsymbol{\delta}_a^{(t)}} \mathcal{L}\big(f_d(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)}), \boldsymbol{x}\big) \\
=&\nabla_{\boldsymbol{\delta}_a^{(t)}} \mathcal{L}\big(f(\text{FTTR}(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)})), \boldsymbol{x}\big) \\
=&\nabla_{\boldsymbol{\delta}_a^{(t)}} \mathcal{L}\big(f(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)} + \boldsymbol{\delta}_d^{(K)}), \boldsymbol{x}\big) \\
=&\frac{\partial \mathcal{L}}{\partial f} \cdot \frac{\partial f}{\partial(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)} + \boldsymbol{\delta}_d^{(K)})} \cdot \frac{\partial(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)} + \boldsymbol{\delta}_d^{(K)})}{\partial \boldsymbol{\delta}_a^{(t)}} \\
=&\frac{\partial \mathcal{L}}{\partial f} \cdot \frac{\partial f}{\partial(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)} + \boldsymbol{\delta}_d^{(K)})} \cdot \big(\boldsymbol{I} + \frac{\partial \boldsymbol{\delta}_d^{(K)}}{\partial \boldsymbol{\delta}_a^{(t)}}\big).
\end{aligned} \tag{34}$$

According to Algorithm 1, $\boldsymbol{\delta}_d^{(K)}$ is obtained through an iterative optimization process with respect to $\boldsymbol{\delta}_a^{(t)}$. In our experiments, $K = 1$, and thus we have:

$$\boldsymbol{\delta}_d^{(1)} = \text{Clip}_{(-s,s)}\big(\boldsymbol{\delta}_d^{(0)} - s \cdot \operatorname{sgn}(\nabla_{\boldsymbol{\delta}_d^{(0)}} \mathcal{L}_{\text{FTTR}}^{(1)})\big). \tag{35}$$

In our experiments, we adopt the straight-through estimation (STE) for $\text{Clip}(\cdot)$ and $\operatorname{sgn}(\cdot)$ by assuming their gradients are identity mappings. Therefore, the partial derivative of $\boldsymbol{\delta}_d^{(1)}$ with respect to $\boldsymbol{\delta}_a^{(t)}$ is:

$$\frac{\partial \boldsymbol{\delta}_d^{(1)}}{\partial \boldsymbol{\delta}_a^{(t)}} = -s \cdot \frac{\partial}{\partial \boldsymbol{\delta}_a^{(t)}} \big(\nabla_{\boldsymbol{\delta}_d^{(0)}} \mathcal{L}_{\text{FTTR}}^{(1)}\big). \tag{36}$$

Equation 36, i.e., the second term inside the last summation parentheses in Equation 34, requires second-order derivatives. This is computationally expensive in practice. We implement the second-order derivatives using the PyTorch framework, where they are internally computed via Hessian-vector products (HVPs). This procedure is commonly referred to as unrolled optimization in previous literature [3]. It represents a strong adaptive white-box attack against FTTR, as it obtains the complete gradient information through the defense module. Although this process is computationally complex, we still evaluate FTTR against this strong adversary as a benchmark in this work.

BPDA. As unrolling incurs substantial computational overhead, we further explore BPDA [4] as a more efficient alternative. Specifically, BPDA approximates the backward pass of FTTR with an identity mapping while preserving its original forward computation. Consequently, the attack gradient is approximated as:

$$\begin{aligned}
&\nabla_{\boldsymbol{\delta}_a^{(t)}} \mathcal{L}\big(f\big(\text{FTTR}\big(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)}\big)\big), \boldsymbol{x}\big) \\
=&\frac{\partial \mathcal{L}}{\partial f} \cdot \frac{\partial f}{\partial \text{FTTR}} \cdot \frac{\partial \text{FTTR}}{\partial\big(\boldsymbol{x} + \boldsymbol{\delta}_a^{(t)}\big)} \\
\approx&\frac{\partial \mathcal{L}}{\partial f} \cdot \frac{\partial f}{\partial \text{FTTR}} \cdot \boldsymbol{I},
\end{aligned} \tag{37}$$

thereby bypassing the expensive second-order derivatives required by unrolled optimization. Although BPDA provides only an approximate gradient, it has been widely adopted for evaluating defenses involving non-differentiable or iterative modules. In this work, we evaluate FTTR against both Unroll and BPDA to provide a comprehensive assessment under adaptive white-box attacks.

Gray-Box Adversary Benchmarks. For gray-box adversary benchmarks, the attacker crafts adversarial examples based solely on the undefended LIC system f, i.e.,

$$\boldsymbol{\delta}_a = A(f, \boldsymbol{x}). \tag{38}$$

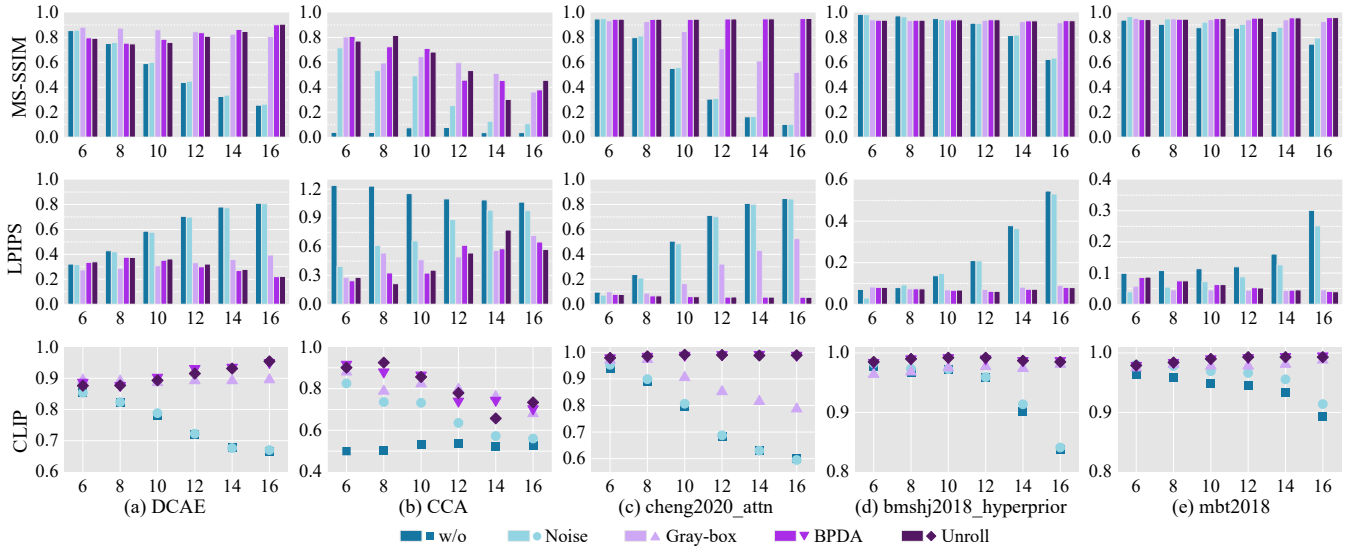

Figure 12: MS-SSIM ↑, LPIPS ↓, and CLIP ↑ of PGD-attacked reconstructions under MSE. X-axis indicates perturbation budgets.

Figure 13: Visualizations of PGD-attacked adversarial examples and their reconstructions under MSE on DCAE. *Example* indicates adversarial examples. *Rec* indicates reconstructions.

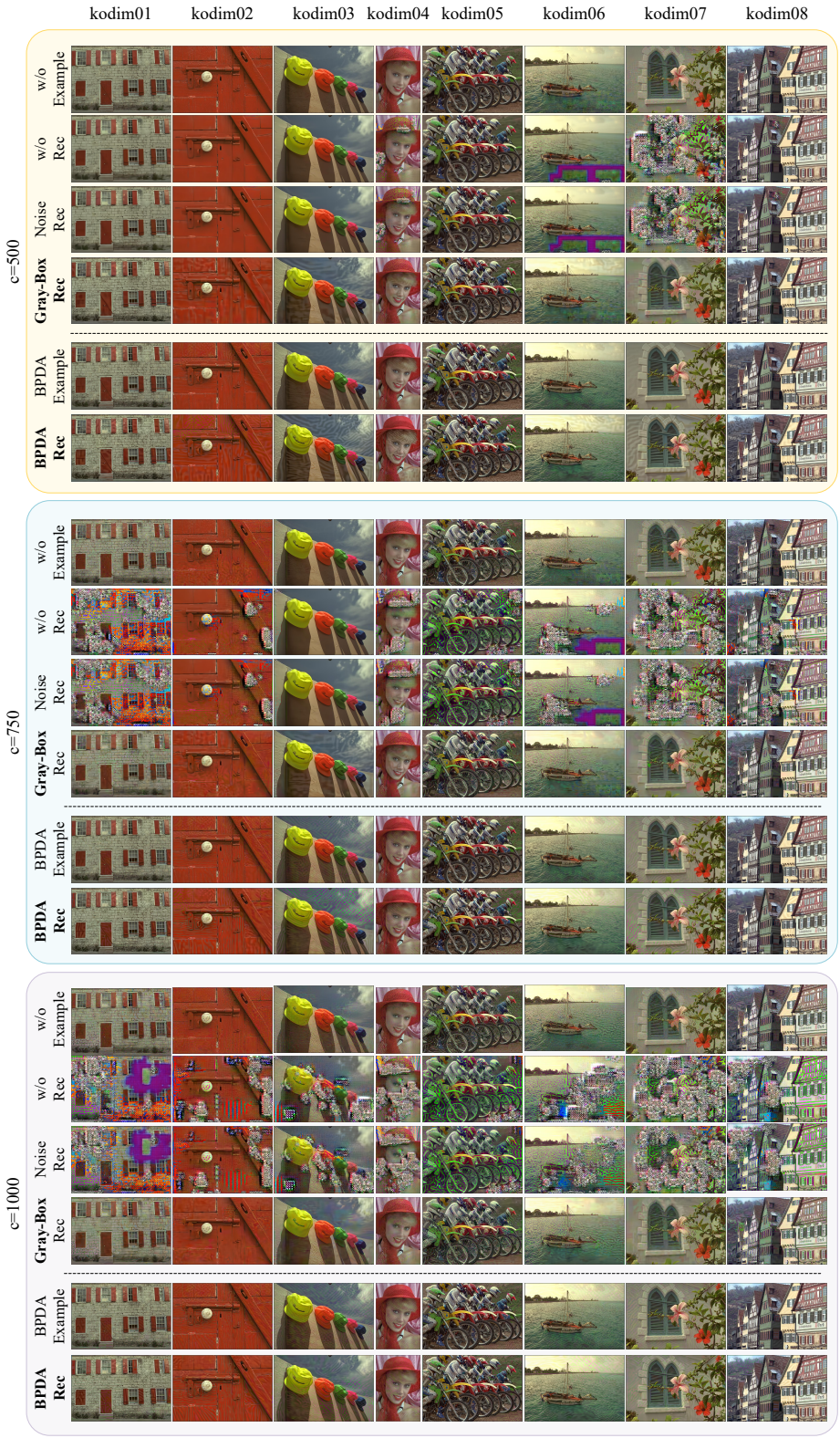

Figure 14: Visualizations of ℓ_2 C&W-attacked adversarial examples and their reconstructions under joint objective of MSE and bpp on DCAE. *Example* indicates adversarial examples. *Rec* indicates reconstructions.