Efficient Adversarial Attack and Training on Learned Image Compression

Jun Kurihara and Heming Sun†*
†Yokohama National University, Kanagawa, Japan
E-mail: kurihara-jun-wc@ynu.jp, hemingsun@ieee.org

***Abstract*—In recent years, Learned Image Compression (LIC) has drawn significant attention because of its powerful coding ability. However, similar to most neural network-based schemes, LIC is vulnerable to adversarial attacks. To mitigate the influence of adversarial attacks, adversarial training is usually adopted to finetune the network. This paper proposes efficient adversarial attack and training methods for LIC, by proposing three losses based on the original image, adversarial image, and reconstructed image. For the attack, we study the effects of three proposed losses on four qualities of classical factorized-prior and hyperprior models. For the adversarial training, all the proposed three losses are used in the finetuning for the three attack scenarios, respectively. We find that using the loss between the adversarial output and the original image achieves strong defense performance against various attacks, improving RD cost by up to 68.6%. Furthermore, we show that updating only the decoding during adversarial training along with reducing the number of iterations, can reduce the training time by up to 82.9% without compromising the defense performance.**

***Index Terms*—Learned Image Compression, Adversarial Attack, Adversarial Training**

I. Introduction

In modern society, the transmission and reception of images have become a part of daily life, and the importance of efficient image compression technologies continues to grow. In response to this demand, learned image compression (LIC) based on neural networks have attracted significant attention in many literature[1]–[5].

However, LIC models are known to be vulnerable to adversarial attacks, where even imperceptible perturbations added to the input image can cause a significant degradation in model output performance [6][7][8]. Therefore, for the practical deployment of LIC models, it is essential to develop effective defense methods against such attacks, and several studies have been proposed in this direction [9].

Zhu et al. [6] evaluate the effectiveness of attacks in different dimensions, including attack methods, models, and targets such as image quality and bitrate. Chen et al. [7] perform attacks that focused on the quality of the reconstructed images using various attack methods, while Liu et al. [8] target the bitrate of the reconstructed images in multiple models. However, none of these studies explore the use of different loss functions for attacks and defenses in LIC models, and the choice of loss function varies from paper to paper.

In this paper, we conduct adversarial attacks on two representative LIC models using the projected gradient descent (PGD) method [10], which is an iterative attack algorithm. We define three distortion-based loss functions and a rate-based one for gradient computation and compare the impact of each on image quality and bitrate degradation. Figure 2 illustrates the generation of adversarial images and their outputs.

(a) Original image (b) Reconstructed image w/ proposed adversarial attack (c) Reconstructed image w/ proposed adversarial training

Fig. 1: (a) Original image, (b) reconstructed image by the proposed PGD attack, and (c) reconstructed image by the proposed efficient adversarial training. Noticeable artifacts in (b) are significantly reduced in (c).

As a defense, we apply adversarial training to the LIC models using the same loss functions as in the attacks. These losses are employed both for generating adversarial images and for computing gradients during model updates. In addition, we introduce a time-efficient training scheme to reduce training time without significantly sacrificing defense performance.

The contributions of this paper are three-fold:

- For the adversarial attack, we explore the effects of three proposed losses on four qualities of classical factorized-prior and hyperprior models. The results show that the proposed method yields an increase in RD cost of 11.81x (36.735 vs. 3.110).
- For the adversarial training, we apply the proposed three losses in each of the three attack scenarios, respectively. We find that keeping the same loss in adversarial training with attack will not always bring the best defense effect. The proposed loss between the adversarial reconstructed image and the original image achieves strongest defense performance, improving RD cost by up to 68.6%.
- To accelerate the adversarial training, we propose to reduce the number of PGD iterations for adversarial image generation from 40 to 5, and only finetune decoder parameters. Results show that up to 82.9% training time is saved with almost the same defense performance.

*Heming Sun is the corresponding author.

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

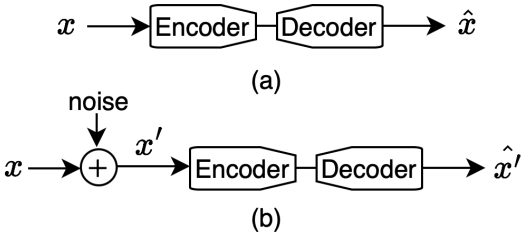


Fig. 2: (a) Coding framework without input perturbation. x is original image, $\hat{x}$ is reconstructed image. (b) Coding framework with input perturbation. x' is polluted image with adversarial noise, $\hat{x}'$ is corresponding reconstructed image. We propose multiple adversarial noises in Section III-A2.

II. Learned Image Compression model

Learned image compression (LIC) is a technique that efficiently compresses and reconstructs images by transforming them into low-dimensional latent representations using convolutional neural networks. Among them, the factorized-prior model [1] and the hyperprior model [2] are fundamental architectures that serve as the basis for many other LIC models. In the factorized-prior model, the input image x is passed through an encoder g_a to produce a latent representation y, which is quantized to $\hat{y}$, entropy coded into a bitstream b_y, and decoded by a decoder g_s to reconstruct the image $\hat{x}$. The hyperprior model extends this structure by introducing a hyper encoder h_a that extracts an additional latent z from y. The latent z is quantized to $\hat{z}$, entropy coded to b_z, and decoded by a hyper decoder h_s to generate side information used to model the distribution of y more accurately. This results in improved compression efficiency over the factorized-prior model.

In these LIC models, a quality parameter controls the trade-off between compression rate and reconstruction fidelity. Lower quality levels achieve higher compression at the expense of image quality, while higher levels preserve more detail with increased bitrate. The models in this paper support eight quality levels, from 1 to 8.

A. Image Evaluation Metrics

To evaluate image compression performance, we use Peak Signal-to-Noise Ratio (PSNR) [dB] and bitrate [bpp]. PSNR quantifies the distortion, while bitrate indicates the size of the compressed data. Since these two are typically in trade-off, we use RD cost as a comprehensive metric, defined as follows.

$$\text{RD cost} = \lambda \cdot 255^2 \cdot 10^{-0.1 \cdot \text{PSNR}(x,\hat{x'})} + \text{Rate} \qquad (1)$$

λ is a parameter that controls the weight between distortion and rate. Following the default configuration provided by CompressAI, we use the following λ values for quality levels 2, 4, 6, and 8: 0.0035, 0.0130, 0.0483, and 0.1800, respectively.

Algorithm 1 PGD Attack with Proposed Losses

Input: original image x, number of steps T, step size α, maximum perturbation ϵ
Output: Adversarial image x'

Initialize $x' = \text{clamp}(x + \mathcal{U}(-\epsilon, \epsilon), 0, 1)$
for $i < T$ **do**
  Compute proposed losses in Eq. (2) **–** (5)
  Compute the gradient of the attack loss: $\frac{\partial Loss}{\partial x'}$
  Generate noise under constraints:
  $delta = \text{clamp}(x' + \alpha \cdot sign(grad) - x, -\epsilon, \epsilon)$
  Generate the adversarial image:
  $x' = \text{clamp}(x + delta, 0, 1)$
  $i = i + 1$
end for
return x'

III. Proposed Method

A. Proposed Attack Method

1) PGD Attack: Projected Gradient Descent (PGD) is a widely used adversarial attack method. Similarly to the originally proposed Iterative Fast Gradient Sign Method (IFGSM), PGD iteratively perturbs the input image x in the direction that maximizes the loss, using model gradients. However, unlike IFGSM, PGD includes a projection step to keep the perturbation within a specified range, preserving visual similarity between the adversarial image x' and the original image x, while still degrading the reconstructed output $\hat{x'}$. These enhancements make PGD more effective and faster than IFGSM. Hence, PGD is also used in adversarial training, offering both better robustness and shorter training times.

The algorithm 1 shows the PGD attack process. PGD starts by adding random noise to the input and then applies iterative updates while keeping perturbations within a defined limit. In this paper, we set the number of steps $T = 40$, step size $\alpha = 0.01$, and maximum perturbation $\epsilon = 0.03$, to maintain the perceptual similarity (PSNR $\geq 30\,\text{dB}$) between x and x'.

Attacks are conducted on both the factorized-prior and hyperprior models at quality 2, 4, 6, and 8. By evaluating attacks in multiple quality settings, we aim to analyze trends in model vulnerability with respect to image quality.

2) Attack Objectives: In adversarial attacks on image compression models, the choice of the loss function plays a central role in defining the attack objective. Depending on the form of the loss, it is possible to target different aspects of the system, such as the visual quality of the reconstructed image (e.g., PSNR) or the bitrate of the compressed data.

We first focus on attacks that aim to degrade the visual quality of reconstructed images. To cause this deterioration, we adopt the Mean Squared Error (MSE) loss function commonly used in distortion attack. The MSE loss is calculated by the distance between two images, and different formulations can be applied depending on which image pairs are compared.

Specifically, we define three types of distortion losses based on combinations of the original image x, the adversarial image

x', and the reconstructed image $\hat{x}$, $\hat{x'}$as follows:

$$\text{Loss}_1 = \text{MSE}(x, \hat{x}') \tag{2}$$

$$\text{Loss}_2 = \text{MSE}(x', \hat{x}') \tag{3}$$

$$\text{Loss}_3 = \text{MSE}(\hat{x}, \hat{x}') \tag{4}$$

These different loss definitions allow for a comparative analysis of how each formulation affects the effectiveness of the attack and the robustness of the target models.

In addition to attacks targeting image quality, we also consider attacks that aim to increase the bitrate, thereby inflating the compressed data size. For bitrate attacks, the loss function is defined as the expected negative log-likelihood of the latent representation y' and the hyperprior latent representation z', based on the entropy model in the hyperprior architecture. Specifically, the bitrate loss is formulated as follows:

$$\text{Loss}_{\text{bpp}} = \mathbb{E}_{y',z'}\left[-\log_2 p_{\hat{y}}(y' \mid \hat{z})\right] + \mathbb{E}_{z'}\left[-\log_2 p_{\hat{z}}(z')\right] \tag{5}$$

$p_{\hat{y}}$ and $p_{\hat{z}}$ denote the learned entropy models for the latent representation y' and z', respectively. Using this loss function during the attack, it becomes possible to increase the bitrate of the compressed output and degrade the compression efficiency.

B. Proposed Defense Method

LIC models are known to be vulnerable to external adversarial attacks. As a countermeasure, adversarial training has been extensively studied. In this approach, LIC model is finetuned using adversarial images in the training dataset, thereby improving its robustness against such attacks. However, training the model exclusively on adversarial images often leads to a significant decline in performance on clean, non-attacked images compared to the original model. To mitigate this issue, we include both adversarially perturbed ("dirty") images and clean images in the training dataset. This mixed-data approach helps preserve high performance even in clean images. In our method, adversarial images are generated during training by iteratively updating the model weights. Since these adversarial images are tailored to the updated model at each training step, the resulting defense is expected to be more effective.

The dirty data used for training are generated based on Algorithm 1. In this paper, we conduct defense experiments against both bitrate attack and quality attacks. For adversarial training against both types of attacks, we employ RDLoss as the loss function to be optimized during training epochs, using the same parameter definitions for the three loss components as those used in the attack experiments to ensure consistency.

For quality attack training, dirty data are generated using the loss functions defined in Equations 2, 3, and 4, and the same parameter settings are applied to RDLoss during training. For both quality and bitrate attack defenses, adversarial training is first conducted using the factorized-prior model and hyperprior model at quality 8. For quality attack defense, we additionally perform adversarial training on both models at qualities 2, 4, and 6 using the loss function that shows the best defense performance at quality 8.

Algorithm 2 Adversarial Training with Proposed Losses

Input: Training dataset $\mathbb{D}$, model f, finetuning iteration N, Encoder parameters θ, Decoder parameters ϕ
Output: Finetuned parameters θ^*, ϕ^*

1: **for** $i = 1$ to N **do**
2: Read clean image x from $\mathbb{D}$
3: Compute clean loss: $\mathcal{L}_{\text{clean}} = \text{RDLoss}(f(x), x)$
4: Generate adversarial image x' by Algorithm 1
// 5-iteration generation if our time-efficient training
5: Compute adversarial loss: $\mathcal{L}_{\text{dirty}} = \text{RDLoss}(f(x'), x)$
// same loss arguments as x' generation
6: Total loss: $\mathcal{L} = \mathcal{L}_{\text{clean}} + \mathcal{L}_{\text{dirty}}$
7: Update θ, ϕ by minimizing $\mathcal{L}$
// update only ϕ if our time-efficient training
8: **end for**
9: **return** θ^*, ϕ^*

To reduce training time, we perform adversarial training with fewer iterations for dirty data generation (from 40 to 5) and update only the g_s network for both models at qualities 2 and 8. This is based on the observation that g_s, being closer to the output, receives stronger gradients during backpropagation, enabling efficient updates with lower computational cost.

We use ImageNet256 [11] as the training dataset and start from pre-trained CompressAI models. Training is performed on 10,000 images for 16 epochs. The learning rate starts at 1e-4 and is reduced to 1e-5 after 10 epochs. With a batch size of 8, the total number of finetuning iterations is 20,000. In Algorithm 2, line 7 shows the computation of the adversarial loss using the same arguments as $Loss_1$ defined in Equation 2. However, in practice, this loss must be identical to the one used during adversarial image generation in line 5.

IV. Experiment Results

For evaluation, we use 24 images from the Kodak dataset for influence and report the mean values of each metric as the results. To quantitatively assess the impact of the attack, following two indicators are used in addition to the RD cost.

$$\Delta\text{bpp} = \frac{\text{bpp}(\hat{x}')}{\text{bpp}(\hat{x})} \tag{6}$$

$$\Delta\ \text{PSNR} = \frac{\text{PSNR}\left(x,\hat{x'}\right) - \text{PSNR}(x,\hat{x})}{\text{PSNR}(x,\hat{x})} \tag{7}$$

In this paper, we visualize the effects of different attack methods on bitrate and PSNR by plotting these two metrics. To evaluate the effectiveness of adversarial training, we compare the RD costs of the pre-trained model and the finetuned model under attack conditions. Let $\text{RDcost}_{\text{ori}}$ denote the RD cost of the pre-trained model and $\text{RDcost}_{\text{ft}}$ that of the finetuned model. The effectiveness of the defense before and after finetuning is evaluated using the following metric.

$$\text{Finetuning}\ \ \text{effect} = \frac{\text{RDcost}_{\text{ft}} - \text{RDcost}_{\text{ori}}}{\text{RDcost}_{\text{ori}}} \tag{8}$$

TABLE I: Evaluation of RD cost under adversarial attacks. Higher RD cost corresponds to more effective attacks.

Target		Distortion Attack			Rate Attack
Model	Quality	Our L_1 [6]	Our L_2	**Our L_3**	Loss_{bpp} [8]
fac	Q2	**0.859**	0.584	0.805	0.675
	Q4	**2.189**	1.272	2.071	1.658
	Q6	**5.667**	2.422	5.641	4.169
	Q8	33.340	11.259	**36.735**	10.401
hyper	Q2	0.704	0.507	0.680	**0.788**
	Q4	1.683	1.000	1.706	**2.140**
	Q6	6.398	2.007	**8.513**	5.899
	Q8	25.706	7.609	**27.530**	13.495
Average		9.568	3.333	**10.460**	4.903

A. Attack Method

Table I shows the RD cost under adversarial attacks targeting both reconstruction quality and bitrate for the factorized-prior and hyperprior models at quality levels 2, 4, 6, and 8. Across all attacks — $Loss_1$, $Loss_2$, $Loss_3$ (for image quality), and $Loss_{bpp}$ (for bitrate) — attack effectiveness increases with model quality. Comparing the two attack types solely in terms of RD cost reveals that attacks targeting reconstruction quality are generally more effective than those targeting bitrate.

For quality attacks, $Loss_2$ is consistently less effective than $Loss_1$ and $Loss_3$, suggesting that it is unsuitable as a distortion loss function for attacks. $Loss_1$, which is used in [6], and $Loss_3$ perform comparably overall. $Loss_1$ is more effective for the factorized-prior model at qualities 2, 4, and 6, and for the hyperprior model at quality 2. Conversely, $Loss_3$ outperforms $Loss_1$ for the factorized-prior model at quality 8 and for the hyperprior model at qualities 4, 6, and 8. This suggests that $Loss_3$-based attacks become more effective as reconstruction quality increases. From the perspective of average RD cost, $Loss_3$ is also the most powerful attack.

For bitrate attacks, effectiveness increases with model quality. The hyperprior model is consistently more vulnerable than the factorized-prior model across all qualities, likely due to its structural difference: While the factorized-prior model utilizes only latent representation y, the hyperprior model introduces an additional latent representation z which introduces more sensitivity to bitrate attacks. Additionally, while [8] uses a 600-step FGSM-based bitrate attack, our 40-step PGD-based attack achieves a higher RD cost of 1.658 (versus 1.61 in their work) on the factorized-prior model at quality 4, demonstrating improved performance with significantly fewer iterations.

Figure 3 illustrates the changes in PSNR and bitrate (Δ PSNR and Δ bpp) caused by adversarial attacks. A larger Δ bpp indicates a greater loss in bitrate, while a more negative Δ PSNR reflects a stronger degradation in reconstruction quality. Hence, points toward the lower right represent stronger overall attack impact. As the $Loss_2$-based attack shows low RD cost (Table I), it is omitted from Figure 3. Each point in the graph represents an attack at quality levels 2, 4, 6, and 8. As seen in Table I, higher quality levels lead to stronger attacks, forming a curve where Δ PSNR decreases with quality.

When comparing attacks using $Loss_1$ and $Loss_3$, the $Loss_3$ curves consistently lie to the right of those for $Loss_1$, indicating a stronger impact on bitrate. In quality 8, $Loss_3$-based attacks also cause greater PSNR degradation. These results suggest that as image quality increases, $Loss_3$ becomes more effective in degrading both image quality and bitrate.

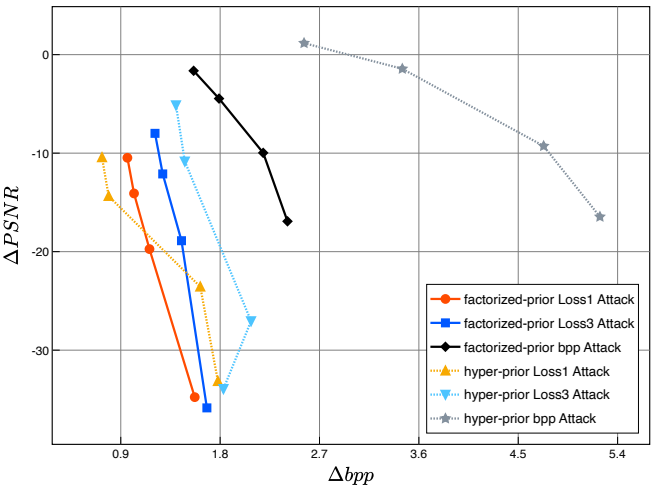
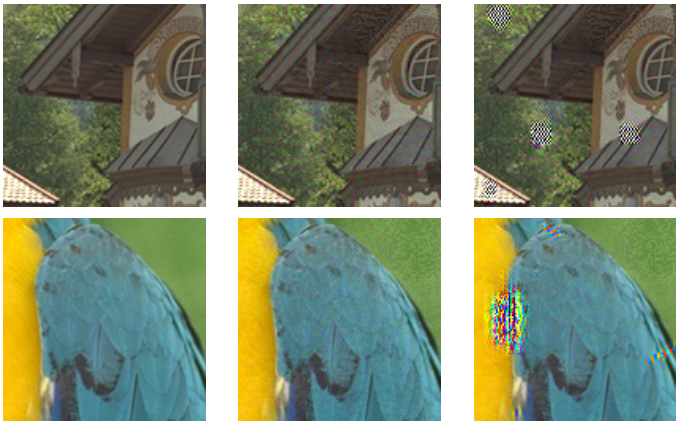


Fig. 3: Effect of adversarial perturbations on PSNR and bitrate. A larger absolute change in PSNR or bitrate (bpp) indicates a stronger adversarial attack. Overall, the hyperprior model exhibits greater bitrate degradation than the factorized-prior model. Although both $Loss_1$ and $Loss_3$ target image quality, $Loss_3$-based attacks cause a notable impact on bitrate.

Next, when comparing the two models under the same loss function, the hyperprior model is generally more vulnerable to bitrate degradation, likely due to structural differences in their latent representations, as discussed earlier. Interestingly, for $Loss_1$ attacks at quality 2 and 4, the factorized-prior model shows greater bitrate degradation. However, in these cases, Δ bpp is less than 1, indicating an improvement in bitrate. This improvement is marginal in the factorized-prior model but more pronounced in the hyperprior model, which appears more sensitive to bitrate degradation.

Based on the results in Table I and Figure 3, we conclude that while the $Loss_1$ and $Loss_3$ attacks exhibit similar overall effectiveness, the superiority of $Loss_3$ becomes more evident as the original image quality increases.

Figure 4 visualizes the adversarial images and their corresponding reconstructions. The top row shows comparisons for the factorized-prior model under quality 8, attacked using $Loss_3$. The bottom row shows comparisons for the hyperprior model under quality 8, attacked using $Loss_1$. In both cases, the adversarial image x' is visually almost indistinguishable from the original image x, but the reconstructed image $\hat{x}'$ exhibits clearly perceptible noise. The factorized-prior model produces multiple regions of black-and-white noise, while the hyperprior model generates large and colorful artifacts. Such artifacts tend to become more prominent in higher qualities.



(a) original x (b) adversary x' (c) output $\hat{x'}$

Fig. 4: Adversarial images and their reconstructions. The top and bottom rows show results for the factorized-prior and hyperprior models, respectively. While no visible difference is observed between the original image x and the adversarial image x', the reconstruction $\hat{x'}$ clearly exhibits noise artifacts.

B. Defense Method

Table II shows the RD cost and defense effectiveness for models finetuned through adversarial training using three different loss functions in factorized-prior and hyperprior models with quality 8. Although adversarial training improves robustness against attacks, it also causes degradation in output performance on clean (w/o attack) inputs. This degradation is likely due to the inclusion of dirty data in the training set.

Among the loss functions, $Loss_3$-based defense is most effective when the attack also uses $Loss_3$, but it performs poorly against other attack types and significantly degrades performance on clean inputs, suggesting that $Loss_3$ is not suitable for general-purpose defense. $Loss_2$-based defense shows moderate effectiveness against most attacks except bitrate-based ones and yields the smallest degradation on clean inputs. In contrast, $Loss_1$-based defense consistently achieves high robustness across all types of attack while maintaining relatively low degradation on clean images. Given the uncertainty of the attack method that might be used in practice, a defense strategy that is broadly effective and preserves clean image quality is desirable. Therefore, adversarial training using $Loss_1$ is considered the most balanced and effective defense.

Figure 5 evaluates the defense effectiveness of adversarial training in terms of bitrate (bpp) and PSNR. As shown in Table II, $Loss_1$ is the most effective loss function in the defense experiment at quality 8. Based on this result, Adversarial training with $Loss_1$ is applied to both the factorized-prior and hyperprior models in quality 2, 4, 6 and 8. For evaluation, the factorized-prior model is attacked using $Loss_1$, and the hyperprior model using $Loss_3$. The figure compares models trained with adversarial training against baselines without it. In each curve, the top point indicates quality 2, followed by quality 4, 6, and 8. Across all settings, adversarial training consistently improves both PSNR and bitrate, with PSNR improvements more pronounced at higher quality levels.

TABLE II: RD cost (adversarial training effect (%)). $Loss_1$ shows the best defense performance among three losses.

(a) factorized-prior, Quality 8

Method	w/o Atk.	Distortion Atk.			Rate Atk.
		Loss_1	Loss_2	Loss_3	Loss_{bpp}
w/o Def.	3.110 (-)	33.340 (-)	11.259 (-)	36.735 (-)	10.401 (-)
L_1 Def.	3.342 (+7.5%)	12.843 **(-61.5%)**	4.083 **(-63.7%)**	11.548 (-68.6%)	5.474 **(-47.4%)**
L_2 Def.	3.197 **(+2.8%)**	14.372 (-56.9%)	4.424 (-60.7%)	14.526 (-60.5%)	11.282 (+8.5%)
L_3 Def.	4.239 (+36.3%)	16.205 (-51.4%)	4.968 (-55.9%)	10.638 **(-71.0%)**	5.873 (-43.5%)

(b) hyperprior, Quality 8

Method	w/o Atk.	Distortion Atk.			Rate Atk.
		Loss_1	Loss_2	Loss_3	Loss_{bpp}
w/o Def.	2.690 (-)	25.706 (-)	7.609 (-)	27.530 (-)	13.495 (-)
L_1 Def.	2.883 (+7.2%)	12.289 **(-52.2%)**	3.776 **(-50.4%)**	11.019 (-60.0%)	4.380 **(-67.5%)**
L_2 Def.	2.715 **(+0.9%)**	13.458 (-47.6%)	4.039 (-46.9%)	13.543 (-50.8%)	12.419 (-8.0%)
L_3 Def.	3.786 (+40.7%)	15.798 (-38.5%)	4.556 (-40.1%)	10.094 **(-63.3%)**	4.998 (-63.0%)

TABLE III: Time-efficient adversarial training. "Time" denotes the total training time (in seconds) for adversarial finetuning.

Target	40 Iter.		Our 5 Iter.		Our 5 Iter. + g_s	
	RD cost	Time	RD cost	Time	RD cost	Time
fac-Q2	0.756	31203	0.760	5921	0.814	5201
fac-Q8	12.843	57818	12.943	11270	13.070	9913
hyper-Q2	0.674	31056	0.688	5890	0.707	5168
hyper-Q8	12.289	58414	12.560	11125	12.753	9857

Figure 6 shows the reconstruction results before and after adversarial training. The top and bottom rows correspond to the factorized-prior and hyperprior models, respectively. For each model, (b) shows the reconstruction of an adversarial input before finetuning, and (c) the reconstruction after finetuning. Noticeable noise artifacts observed before adversarial training are visually reduced afterward, demonstrating the effectiveness of adversarial training in improving reconstruction quality.

Table III summarizes the results of time-efficient training under attack and defense based on $Loss_1$. For the factorized-prior model at quality 8, conventional adversarial training takes 57,818 seconds with an RD cost of 12.843. By reducing the number of iterations for dirty data generation from 40 to 5, the training time is significantly reduced. Furthermore, by restricting the trainable parameters to the g_s network, the time is further reduced to 9,913 seconds, with only a slight increase in the RD cost to 13.070. In efficient training, a slight sacrifice in PSNR leads to an improved bitrate, resulting in RD cost that is almost equivalent to that of conventional training. This approach reduces training time by 82.9% while maintaining defense performance, and shows consistent effectiveness across both models in qualities 2 and 8.

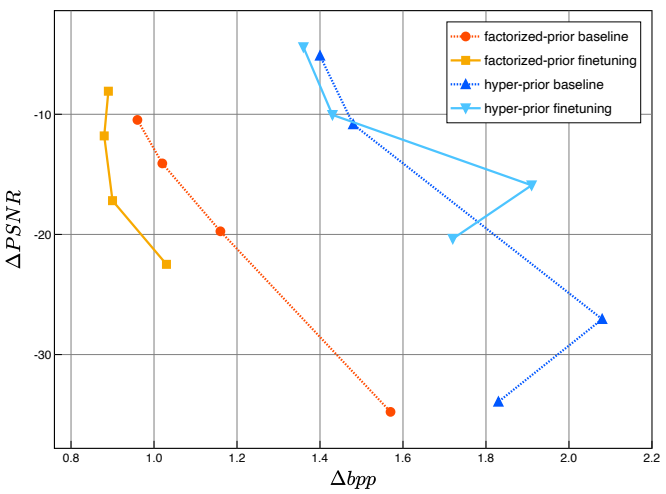
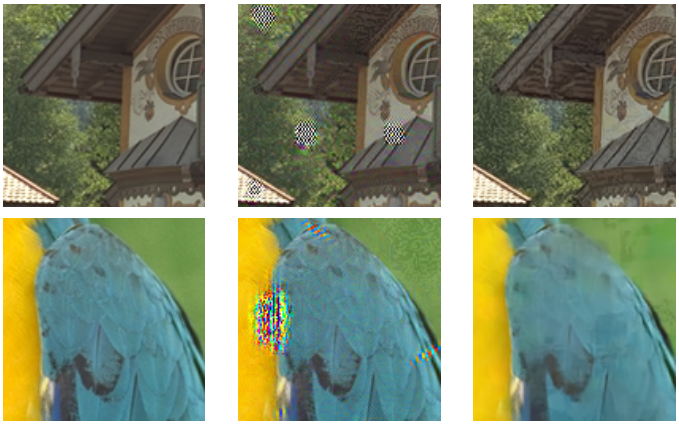


Fig. 5: Improvements in bitrate and PSNR through adversarial training. Adversarial training with $Loss_1$ improves both PSNR and bitrate compared to the baseline without adversarial training, across all quality levels (2, 4, 6, and 8).



(a) original x (b) w/o defense $\hat{x}'$ (c) w/ defense $\hat{x}'$

Fig. 6: Improvement in reconstruction quality through adversarial training. (b) shows the reconstruction $\hat{x}'$ from an adversarial input before training, and (c) shows the result after training, where noise artifacts are visibly reduced.

V. Conclusions

This paper investigates how loss function definitions affect adversarial attacks and adversarial training in two LIC models: the factorized-prior and hyperprior models. In attack experiments, we find that using the loss between $\hat{x}$ and $\hat{x}'$ significantly degrades both PSNR and bitrate, increasing the RD cost by up to 36.735, with visually noticeable noise in $\hat{x}'$. For defense, adversarial training using the loss between x and $\hat{x}'$ improves robustness by up to 68.6% and reduces the visible noise observed in $\hat{x}'$. We also demonstrate a time-efficient training scheme that reduces the training time by up to 82.9% through fewer attack iterations and updating only the g_s network. Although we use the same loss for both generating adversarial images and optimizing the model during training, our results indicate that the optimal loss may differ between attack and defense. Future work should investigate whether using separate losses for dirty data generation and model optimization can further enhance robustness.

VI. Acknowledgment

This work was supported in part by JSPS KAKENHI Grant Number JP23K16861, in part by Hoso Bunka Foundation, in part by Telecommunications Advancement Foundation, and in part by SCAT.

References

[1] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[2] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018.

[3] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7939–7948.

[4] Y.-H. Ho, C.-C. Chan, W.-H. Peng, H.-M. Hang, and M. Domański, "Anfic: Image compression using augmented normalizing flows," *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 613–626, 2021.

[5] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, "End-to-end learnt image compression via non-local attention optimization and improved context modeling," *IEEE Transactions on Image Processing*, vol. 30, pp. 3179–3191, 2021.

[6] T. Zhu, H. Sun, X. Xiong, X. Zhu, Y. Gong, Y. Fan, *et al.*, "Attack and defense analysis of learned image compression," *arXiv preprint arXiv:2401.10345*, 2024.

[7] T. Chen and Z. Ma, "Toward robust neural image compression: Adversarial attack and model finetuning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7842–7856, 2023.

[8] K. Liu, D. Wu, Y. Wu, *et al.*, "Manipulation attacks on learned image compression," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 6, pp. 3083–3097, 2023.

[9] M. Song, J. Choi, and B. Han, "A training-free defense framework for robust learned image compression," *arXiv preprint arXiv:2401.11902*, 2024.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[11] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.