Denial-of-Service Attacks on Learned Image Compression

Kang Liu¹, Di Wu¹, Yiru Wang², Dan Feng¹, Benjamin Tan³, Siddharth Garg⁴
¹Huazhong University of Science and Technology
²Beijing University of Posts and Telecommunications
³University of Calgary, ⁴New York University
¹{kangliu, diwu1294, dfeng}@hust.edu.cn, ²yiruwang@bupt.edu.cn
³benjamin.tan1@ucalgary.ca, ⁴siddharth.garg@nyu.edu

Abstract

Deep learning techniques have shown promising results in image compression, with competitive bitrate and image reconstruction quality from compressed latent. However, while image compression has progressed towards higher peak signal-to-noise ratio (PSNR) and fewer bits per pixel (bpp), their robustness to corner-case images has never received deliberation. In this work, we, for the first time, investigate the robustness of image compression systems where imperceptible perturbation of input images can precipitate a significant increase in the bitrate of their compressed latent. To characterize the robustness of state-of-the-art learned image compression, we mount white and black-box attacks. Our results on several image compression models with various bitrate qualities show that they are surprisingly fragile, where the white-box attack achieves up to 56.326× and black-box 1.947× bpp change. To improve robustness, we propose a novel model which incorporates attention modules and a basic factorized entropy model, resulting in a promising trade-off between the PSNR/bpp ratio and robustness to adversarial attacks that surpasses existing learned image compressors.

1 Introduction

Image compression is a core task in the image processing pipeline, and can substantially reduce both local storage resources or bandwidth requirements if images are transmitted to remote servers. Conventional image compression methods [20, 22, 25, 28] (e.g., JPEG2K [22]) rely on hand-crafted lossy compression followed by entropy coding. Recently, deep learning (DL)-based compression methods [1, 2, 6, 15, 17, 18, 27] have demonstrated superior performance compared to hand-crafted techniques, and are now the state-of-the-art in terms of rate-distortion trade-off.

As learned image compression transition to practice via standardization, examination of their robustness of adversarial perturbation is a key question given the notorious susceptibility of deep learning to imperceptible input modification. How do these image compressors perform under corner-case (adversarial) conditions? In this paper, we introduce and investigate bitrate robustness as a new and important metric in the development of learned image compression. We frame this study within an adversarial context: an adversary seeks to imperceptibly modify inputs to drastically increase the compression bitrate without impacting the quality of reconstruction (i.e., distortion). The adversary can then launch denial-of-service attacks, exhausting local storage resources or transmit bandwidth.

We make the following contributions (see also Figure 1): (1) the first comprehensive evaluation of the bitrate robustness of state-of-the-art learned image compressors; (2) novel white-box and black-box attack formulations that craft adversarial images with increased storage requirements for corresponding compressed latent; (3) formulation and evaluation of a novel network architecture
incorporating a factorized entropy prior and attention modules that considers robustness; and (4) insights on the entropy models and network components regarding their robustness implications.

2 Preliminaries and threat model

2.1 Learned image compression

We adopt similar notation as in prior work \cite{2, 6, 18}. Recent work adopts transform coding \cite{10} (as shown in Equation 1) for learned image compression, where image $x$ is mapped into compressed latent $y$ by analysis transform $g_a$ (a deep network). The latent is quantized as $\hat{y}$ and entropy coded. For reconstruction, compressed latent $\hat{y}$ (obtained from entropy decoding) is passed through synthesis transform $g_s$ to yield $\hat{x}$. The weights for the analysis and synthesis transforms are denoted by $\theta_{g_a}$ and $\theta_{g_s}$, respectively. Mathematically, we can write:

$$y = g_a(x; \theta_{g_a}), \quad \hat{y} = Q(y), \quad \hat{x} = g_s(\hat{y}; \theta_{g_s}), \quad \text{bpp}(x) = \frac{\text{str_len(arith_coder}(\hat{y}))}{\text{# of total pixels}}$$ (1)

A key challenge in learned image compression is estimating the entropy of quantized latent $\hat{y}$. To this end, learned image compression methods use an entropy model, which is a prior on $\hat{y}$. Learned image compression techniques differ, in large part, in how the entropy model is constructed; typically, more accurate models result in improved performance. Our attacks target the entropy model. Further, we show, the choice of entropy model also has a large impact on robustness. Since these are key to our study, we describe the entropy models proposed in prior work in some detail.

The “factor” model. The simplest entropy model is just a fully factorized model \cite{1}, as shown in Equation 2. Here $\psi$ collectively denotes the parameters of each univariate distribution $p_{y_i | \psi}(\psi)$. We refer to image compression with a factorized prior as “factor” \cite{1}. Specifically, during training, the quantization of $y$ is approximated by adding uniform noise $\mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ to generate $\hat{y}$; at inference, integer rounding is used instead. To ensure a better match of the prior (of quantized latent $\hat{y}$) to the marginal (of continuous-valued latent $y$), we convolve each non-parametric density with a standard uniform distribution to model each $\hat{y}_i$ as illustrated by Equation 2.

$$p_{\hat{y} | \psi}(\hat{y} | \psi) = \prod_i (p_{y_i | \psi}(\psi) \ast \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\hat{y}_i)$$ (2)

\footnote{modeled by a 5-layer factorized entropy network in practice}
With the factor model in place, the training goal is to minimize the weighted sum of the rate and distortion, using a Lagrangian multiplier $\lambda$ to control the rate-distortion trade-off, as shown in Equation (3) below.

$$
L = R(\hat{y}) + \lambda \cdot D(x, \hat{x}) = \mathbb{E}[-\log_2(p_{\hat{y}|\psi}(\hat{y} | \psi))] + \lambda \cdot D(x, \hat{x})
$$

Hierarchical hyperprior models. As shown by [2], strong spatial dependencies remain among the elements of $y$. Hierarchical entropy models seek to exploit structure information in the compressed latent $y$, improving compression performance. In Equation (4) $h_a$ and $h_s$ denote the analysis and synthesis transform of the hyperprior entropy model, each parameterized by $\theta_{h_a}$ and $\theta_{h_s}$. Here $p_{\hat{y}|z}(\hat{y}|z)$ is the estimated distribution of $\hat{y}$ conditioned on $z$, often called the side information. In hyperprior models, $z$ is also arithmetically coding along with $\hat{y}$, and included in the compressed bitstream. The modeling of $z$ itself still uses a non-parametric fully factorized model such that $p_{z|\psi}(z|\psi) = \prod_i (p_{z_i|\psi}(z_i) \ast \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\hat{z}_i)$. The training loss of image compression with a hyperprior entropy model is given by Equation (5).

$$
y = g_a(x; \theta_{g_a}), \quad z = h_a(y; \theta_{h_a}), \quad \hat{y} = Q(y), \quad \hat{z} = Q(z), \quad \hat{x} = g_s(\hat{y}; \theta_{g_s}), \\
p_{\hat{y}|z}(\hat{y}|z) \leftarrow h_s(\hat{y}; \theta_{h_s}), \quad \text{bpp}(x) = \frac{\text{str}_{\text{len}}(\text{arith}_c\text{oder}(\hat{y})) + \text{str}_{\text{len}}(\text{arith}_c\text{oder}(\hat{z}))}{\# \text{ of total pixels}}
$$

$$
L = R(\hat{y}) + R(\hat{z}) + \lambda \cdot D(x, \hat{x}) \\
= \mathbb{E}[-\log_2(p_{\hat{y}|z}(\hat{y} | z))] + \mathbb{E}[-\log_2(p_{z|\psi}(z | \psi))] + \lambda \cdot D(x, \hat{x})
$$

In prior work, $p_{\hat{y}|z}(\hat{y}|z)$ is modeled as a Gaussian distribution with mean $\mu$ and scale $\sigma$; each element $\hat{y}_i$ has its own $\mu_i$ and $\sigma_i$, which are returned by $h_s$ as shown in Equation (6). We denote such image compression models as “hyperPri” [18]. In cases where only zero-mean Gaussian ($\mu = 0$) is used, we refer to these compression models as “hyper0” [2].

$$
p_{\hat{y}|z}(\hat{y}_i|z) \sim \mathcal{N}(\mu, \sigma^2), \quad \mu, \sigma = h_s(\hat{z}; \theta_{h_s})
$$

Context models. An even more accurate entropy model can be developed by predicting $\mu$ and $\sigma$ conditioned on both $\hat{z}$ and the causal context of all previously decoded $\hat{y}_i$, as expressed in Equation (7). In this case, an autoregressive context model (denoted as $f_{cm}$ with parameters $\theta_{cm}$) and a hyperprior $h_s$ are jointly utilized. An entropy parameter network (denoted as $f_{cm}$ with parameters $\theta_{cm}$) combines these two sources of information and generates $\mu_i$ and $\sigma_i$ for each compressed latent $\hat{y}_i$, as shown below:

$$
\phi = h_s(\hat{z}; \theta_{h_s}), \quad \varphi_i = f_{cm}(\hat{y}_{<i}; \theta_{cm}), \quad \mu_i, \sigma_i = f_{cm}(\phi, \varphi_i, \theta_{cm})
$$

We denote image compression models that use this joint autoregressive and hierarchical entropy model as “hyperCm” [15][18].

The baseline “hyperCm” models have been further improved by using residual blocks in the analysis and synthesis transforms; we call this the “residOrg” model. A further enhancement that uses both residual and attention modules in the analysis and synthesis transforms is referred to as “residAttn” model [6]. Note that “residOrg” and “residAttn” both use the same hyperior context model as hyperCm, and differ only in the structure of the analysis and synthesis transforms.

2.2 Threat model

For our experimental evaluation of robustness, we adopt an adversarial angle. The adversary aims to stealthily sabotage the edge device where image compression is used, forcing wasted storage or network bandwidth until resources are exhausted and service is denied (i.e., a denial-of-service (DoS) attack). The adversary’s goal is to craft adversarial images that significantly consume more storage

---

[2]: A limited context (5 × 5 convolution kernels) with masked convolution is used in practice.
space than expected after compression by making the smallest possible change to the input that causes the greatest increase in the bit string length of the compressed latent.

\[
\max \text{ bpp}(x') \text{ s.t. } \|x' - x\| \leq \epsilon
\]  

The adversary’s capabilities are defined by the amount of information they possess for the attack. In the context of attacks against DL, this could include knowledge about the targeted NN’s architecture and weights, training algorithms, training dataset, etc. Here, we explore two attacks: (1) white-box, where the adversary has full access to the learned image compressor’s architecture, weights, and biases; and (2) black-box, where the adversary possesses no architectural information about the target model but can query it with any input. The adversary can train and use substitute NNs for transfer attacks [21] as needed.

3 Proposed attacks and experimental results

3.1 Experimental setup

We evaluate six image compression models from prior work mentioned in Section 2.1: \texttt{factor}, \texttt{hyper0}, \texttt{hyperPri}, \texttt{hyperCm}, \texttt{residOrg}, and \texttt{residAttn}, and our proposed model \texttt{factorAttn}. For the six baseline models, we use pre-trained weights from [3] corresponding to six different reconstruction qualities ranging from quality = 1 (lowest quality) to quality = 6 (highest quality). Each quality level corresponds to a different average bpp optimized using different \(\lambda\) values in training (as tabulated in the Appendix). Our proposed \texttt{factorAttn} model is fine-tuned from the pre-trained models; thus overall we evaluate 42 different models. We also investigate the robustness of JPEG compression with five different bitrate qualities and show the results in the Appendix.

Our experimental evaluation focuses on bpp change and changes to the peak signal-to-noise ratio (PSNR change), defined as follows:

\[
\text{bpp change} = \frac{\text{bpp}(x')}{\text{bpp}(x)}, \quad \text{PSNR change} = \frac{\text{PSNR}(x, \hat{x'})}{\text{PSNR}(x, \hat{x})}
\]  

A positive bpp change reflects an increase in bpp relative to the original bpp, i.e., more bits are required to represent each pixel (corresponding to less efficient compression). PSNR change represents the additional distortion introduced in reconstruction due to perturbation added to the original images; more-negative values indicate larger distortion. In the following discussion, results are reported based on the mean bpp change and PSNR change after perturbing the 24 images of the publicly available Kodak dataset [13]. In all experiments, we used an \(\epsilon = 8/255\) averaged over all pixels [7].

3.2 White-box attack

Since arithmetic coders are near-optimal entropy coders, the entropy of the quantized latent is a good estimation of the length of its compressed bitstream (this is how learned compressors are trained end-to-end). Instead of directly optimizing the actual non-differential bit length, the attack aims to increase the entropy estimation as it approximates the actual bpp (Equation 10). We define the attack loss function \(L_{\text{atk}}(x')\) as the total entropy estimation of \(\hat{y}\) and \(\hat{z}\) (\(\hat{y}\) only in \texttt{factor} models), as shown in Equation 11. Here, \(x'\) are the adversarial images, and \(\hat{y} = Q(g_a(x'; \theta_{g_a}))\) and \(\hat{z} = Q(h_a(g_o(x'; \theta_{g_o}); \theta_{h_a}))\) are the quantized latents for arithmetic coding.

\[
\text{bpp}(x') = \frac{\text{str}_\text{len}(\text{arith_coder}(\hat{y}')) + \text{str}_\text{len}(\text{arith_coder}(\hat{z}'))}{\# \text{ of total pixels}} \approx \mathcal{R}(\hat{y}') + \mathcal{R}(\hat{z}')
\]  

Let \(L_{\text{atk}}(x') = \mathcal{R}(\hat{y}') + \mathcal{R}(\hat{z}') = \mathbb{E}[-\log_2(p_{\hat{y}'|z'}(\hat{y}'))] + \mathbb{E}[-\log_2(p_{\hat{z}'|y'}(\hat{z}')|\psi')]
\]  

In the white-box attack, we adopt a gradient-guided approach to generate adversarial images, inspired by the fast gradient sign approach [9, 14]. The adversary calculates the gradient of the attack loss function for the input image (line 4 in Algorithm 1) and then uses the sign of the gradient \(\text{grad}\) to modify each pixel by step size \(\delta\) in the direction of the gradient. The adversary aims to increase the attack loss by iterative perturbation and the total perturbation after each iteration is bounded by \(\epsilon\) (line 5 in Algorithm 1). During the attack, adversarial images are always subject to range clips to
ensure legitimacy (line 6 in Algorithm 1). The algorithm stops when the attack process has reached the maximum allowed iteration $T$. Attacks on various model architectures share the same attack method, except that different network components are applied when running network inference for entropy estimation. The detailed algorithm of the white-box attack is shown in Algorithm 1.

**Algorithm 1** White-box attack

1: Input: original image $x_0$, attack objective function $L_{atk}$, max perturbation $\epsilon$, step size $\delta$, max iteration $T$
2: Let $x' = x_0$, $t = 0$
3: while $t < T$ do
4: Compute gradients of the attack objective function w.r.t. input image: $\text{grad} = \frac{\partial L_{atk}(x')}{\partial x'}$
5: Obtain image perturbation under constraint: $\text{pert} = \text{clip}_\text{by}_\text{norm}(x' + \delta \cdot \text{sign}(\text{grad}) - x_0, \epsilon)$
6: Obtain adversarial image: $x' = \text{clip}_\text{by}_\text{value}(x_0 + \text{pert}, 0, 1)$
7: $t = t + 1$
8: Return: adversarial image $x'$

**Experimental results** Table 1 summarizes the white-box attack success with respect to bpp change (recall that bpp change is ratio between the bpp of the adversarially perturbed and original images). More detailed results are in the Appendix. Note that larger values correspond to more successful attacks. We observe that the largest increase of bpp (i.e., a substantial $56 \times$ increase in bpp) occurred when attacking the hyperCm model with quality = 1. The factor model, while being the “simplest”, exhibited the least increase in bpp across all quality settings which suggests that the use of hyperprior elements degrades robustness to adversarial inputs. The addition of attention in residAtn compared to residOrg results in lower increases in bpp, across all quality levels. Overall, we find that the hyperCm model is the least robust for the three lowest qualities, hyperCm for intermediate quality and hyperPri for highest qualities.

| Quality   | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn |
|-----------|--------|--------|----------|---------|----------|----------|
| quality = 1 | 1.617  | 4.475  | 4.747    | 56.326  | 5.982    | 2.826    |
| quality = 2 | 1.706  | 3.956  | 3.512    | 32.166  | 7.604    | 3.177    |
| quality = 3 | 1.830  | 5.636  | 7.038    | 24.140  | 9.895    | 3.590    |
| quality = 4 | 2.035  | 5.267  | 8.909    | 24.497  | 30.129   | 5.515    |
| quality = 5 | 2.230  | 5.476  | 32.744   | 22.649  | 25.450   | 7.240    |
| quality = 6 | 2.541  | 6.556  | 21.004   | 17.065  | 19.827   | 8.617    |

Table 2 summarizes the impact on PSNR. As bitrate quality increases, PSNR change gets larger. In most cases, there is minimal image quality loss as shown by PSNR for reconstructed adversarial images, compared to the reconstruction of original images. There is one outlier of the residOrg model with quality = 6, which we discuss further in Section 5.

**3.3 Black-box attack**

In the black-box setting no gradient information is accessible. The adversary can query the compression model iteratively with numerous potentially adversarial images and identify images that result in the greatest increase in bpp. As an exhaustive query using all possible perturbations is infeasible, a common method for black-box attacks is transferring attacks [21], in which adversarial images generated from a substitute network remain adversarial to other black-box networks. Inspired by JPEG compression, which uses the discrete cosine transform (DCT) before quantization and entropy coding, we design a JPEG-like NN as a substitute model for adversarial image generation. As shown in [16], the summation and element-wise product of DCT can be realized directly as a convolution layer of the NN. The application of the quantization table in JPEG is a simple matrix multiplication.
Table 2: Summary of PSNR change in white-box attacks on different model architectures and qualities.

| Quality | factor | hyper0 | hyperPri | hyperCm | residOrg | residAttn |
|---------|--------|--------|----------|---------|----------|-----------|
| 1       | -2.0%  | 0.9%   | -0.6%    | 0.2%    | -1.4%    | -1.0%     |
| 2       | -3.2%  | 0.3%   | -0.8%    | 0.6%    | -2.9%    | -2.9%     |
| 3       | -5.1%  | -0.5%  | -2.2%    | -0.9%   | -5.1%    | -5.6%     |
| 4       | -6.7%  | -2.7%  | -4.2%    | -3.1%   | -14.7%   | -7.5%     |
| 5       | -17.0% | -7.6%  | -6.7%    | -8.1%   | -13.2%   | -11.0%    |
| 6       | -13.6% | -11.4% | -16.0%   | -13.6%  | -43.0%   | -15.2%    |

Entropy estimation is viable through factorized entropy network. Thus, in black-box attack we propose a DCT-Net (shown in Fig. 2), which comprises a DCT convolution layer, a Quantization layer, and a factorized entropy network.

Figure 2: DCT-Net architecture for black-box transferring attacks with DCT implemented as a convolutional layer

Since the DCT computation operates on $8 \times 8$ sub-blocks of each image, the DCT convolution layer will have filters of size $(8, 8)$ and strides of 8, resulting in 192 channels. DCT-Net uses fixed-weight DCT/IDCT convolutional layers; only the 5-layer factorized entropy network is optimized. It allows quick training until convergence.

The black-box attack involves training $N$ instances of substitute DCT-Nets, each using a different quantization table $Q$ as in JPEG compression. $N$ is an adversary-defined parameter. The adversary generates images with the white-box method (Algorithm 1) for each DCT-Net. To evaluate robustness, we see if the adversarial images transfer to the target models. Theoretically, the more DCT-Nets (with different bitrate quality) are trained, the more likely it is for an adversary to find adversarial images that will increase the bpp. In these experiments, we train $N = 5$ DCT-Nets, each with $Q = 10, 30, 50, 70, 90$, respectively.

Experimental results Table 3 summarizes the highest achieved bpp change and Table 4 presents the corresponding PSNR change. The highest attack success achieved in the black-box setting occurs when attacking the hyperCm model with a quality of 6 using DCT-Net with $Q = 90$, where the adversarial images consume $1.94 \times$ more bits than their original ones. As observed in the white-box setting, factor models show better robustness across all bitrate qualities compared to the other models. Models with higher bitrate quality tend to be less robust and, in most cases, exhibit larger PSNR change. As models’ bitrate quality increases, the best attack usually comes from adversarial images transferred by DCT-Nets with higher quality $Q$, which suggests that closer resemblance of different models in bitrate can have higher transferring attack success. We also black-box attack JPEG compression with different bitrate quality and present the results in the Appendix.

4 Towards a more robust image compression model

Our experimental results (Table 1) show that the added attention modules in the residual-block-based analysis/synthesis transform reduces attack success, suggesting its use for robustness enhancement.
Table 3: Summary of highest achieved bpp change in black-box attacks on different models/qualities. Parenthetical numbers indicate the quantization table Q of the substitute DCT-Net, from which the most successful adversarial image was generated.

| quality | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn |
|---------|--------|--------|----------|---------|----------|----------|
| = 1     | 1.157 (10) | 1.319 (10) | 1.347 (10) | 1.382 (10) | 1.414 (10) | 1.415 (10) |
| = 2     | 1.228 (10) | 1.394 (10) | 1.422 (10) | 1.451 (10) | 1.480 (10) | 1.481 (10) |
| = 3     | 1.299 (10) | 1.444 (30) | 1.466 (30) | 1.495 (30) | 1.501 (10) | 1.498 (10) |
| = 4     | 1.352 (30) | 1.545 (50) | 1.549 (50) | 1.582 (30) | 1.614 (30) | 1.611 (30) |
| = 5     | 1.422 (30) | 1.612 (70) | 1.744 (90) | 1.746 (70) | 1.730 (70) | 1.715 (70) |
| = 6     | 1.548 (70) | 1.901 (90) | 1.931 (90) | 1.947 (90) | 1.806 (70) | 1.796 (70) |

Table 4: Summary of PSNR change in black-box attacks on different model architectures and qualities.

| quality | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn |
|---------|--------|--------|----------|---------|----------|----------|
| = 1     | 0.7%   | 1.7%   | 1.7%     | 1.2%    | 1.7%     | 1.5%     |
| = 2     | 0.2%   | 1.2%   | 1.6%     | 1.5%    | 0.6%     | 0.9%     |
| = 3     | -1.0%  | 0.8%   | 1.1%     | 0.8%    | -1.7%    | -1.6%    |
| = 4     | -1.8%  | -1.3%  | -1.1%    | -2.8%   | -3.5%    | -3.8%    |
| = 5     | -4.9%  | -5.3%  | -2.6%    | -5.8%   | -6.3%    | -5.9%    |
| = 6     | -6.8%  | -10.1% | -9.6%    | -9.8%   | -10.9%   | -10.6%   |

Thus, we propose a factorAttn model comprising analysis and synthesis transform with attention modules, accompanied by a basic factorized entropy network. We incorporate a simplified version of the attention module as used in [6] with pre-trained factor models and employ fine-tuning for the sake of training efficiency. We provide more insights on the enhancement of robustness introduced by attention modules in Section 5.
Appendix.

Experimental results Image compression models exhibit similar bitrate if within the same quality group; thus, PSNR/bpp ratio is a reasonable metric representing their compression performance in cases of non-adversarial images. We compare our proposed factorAttn model with the existing models, as before, in both white-box and black-box settings, regarding their robustness and PSNR/bpp performance, as shown in Fig. 3 and Fig. 4, respectively. A larger PSNR/bpp ratio and lower bpp change represent better compression performance and higher robustness, i.e., models placed at the bottom right corners in Fig. 3 and Fig. 4. Our proposed factorAttn model (denoted by G) works the best regarding both PSNR/bpp ratio and robustness for quality = 1 to 4 in both white-box and black-box settings. In higher qualities (5–6), a trade-off appears between PSNR/bpp ratio and robustness, but the factorAttn is still on the Pareto boundary of this trade-off. In addition to the robustness enhancement, attention modules also improve PSNR/bpp ratio when comparing factor and factorAttn models across all bitrate qualities. We attribute this phenomenon to the more accurate entropy estimation of attention architecture [6]. We will further discuss why the simplest factorized entropy model and attention modules work best regarding robustness in Section 5. Detailed results on bpp change and PSNR change of factorAttn models in both attack settings can be found in the Appendix.

5 Discussion

More accurate entropy estimation, less robustness. Our experimental results (Fig. 3 and Fig. 4) on the bpp change of models factor, hyper0, hyperPri, and hyperCm across various bitrate qualities and attack scenarios characterize their different robustness. In most cases, complex entropy models (such as hyper0, hyperPri, and hyperCm) that more accurately estimate entropy (producing higher PSNR/bpp ratio), are more vulnerable to adversarial images. Essentially, the more “accurate” entropy estimation makes strong assumptions on the distribution of non-adversarial images. The less effective entropy modeling network, such as factor, exhibits higher robustness than those more powerful entropy models because the attack process operates by increasing the entropy estimation returned by entropy models. The less accurate the estimation from the entropy model, the lower the attack success, and therefore, higher robustness.

Reconstruction distortion. Models of the various architectures exhibit larger PSNR changes for adversarial images for higher quality levels (Section 3), as those with higher qualities keep more information in compression (use more bits). In adversarial images, bounded by the same perturbation constraints, the perturbations tend to “survive” more after compression with higher quality, resulting
in a larger difference to their originals and, therefore, more reconstruction distortion. Typically, adversarial images introduce negligible distortion as shown by the PSNR change reported in Section 3 and via visual examination of their reconstruction. However, we observed one outlier, the residOrg model with quality = 6, produces a 43% drop in PSNR (Table 2) in the white-box setting. Since we do not put constraints on the reconstruction quality of adversarial images in the attack loss function, minimum distortion in the reconstruction is not guaranteed; large distortion or visible noise could appear. We present more details on the original, adversarial, and reconstructed images of the residOrg models in the Appendix.

Attention enhances robustness. Attention modules can help NNs focus on challenging parts of an image and capture the features of subtle perturbation, allowing for a more accurate entropy estimation against adversarial images by the nature of its construction and training. This is exhibited in our results (Table 1), specifically when we compare model residOrg and model residAtn across all the bitrate qualities in the white-box setting. Prior work [30] highlights the benefit of using attention modules to strengthen image classifiers against adversarial perturbation, which aligns with our observations in image compression.

JPEG robustness. How do learned compressors compare to conventional JPEG? We ran black-box attacks on JPEG compression with different bitrate qualities. Although exhibiting lower PSNR/bpp performance compared to the state-of-the-art learned image compressors, JPEG is generally more robust to adversarial images in the black-box attack, as shown in Fig. 4(e) and Fig. 4(f). More results on black-box attacks on JPEG compressors are detailed in the Appendix.

Robustness to random noise. NN often reliably classify images with noise corruption better than adversarial examples. We observed similarly in learned compression: there was very little change to bpp for inputs with added noise. We provide more results on bpp change after adding random Gaussian noise to a range of model architectures and bitrate qualities, detailed in the Appendix.

Limitations. The image compression models we investigated share a similar underlying autoencoder architecture. While studied widely for image compression, they are not the only architectures available. For example, recurrent neural networks (RNNs) [12, 27] and generative adversarial networks (GANs) [17] can be used for image compression. We did not investigate the effect of network capacity (i.e., the number of layers and number of channels) on robustness, as we were interested in comparing robustness of different entropy models and network components. Future work includes applying our attacks on models with higher bitrate quality, using a different distortion metric for training (e.g., MS-SSIM), and using other datasets.

6 Related work

In addition to prior work [1, 2, 6, 18] that shares an autoencoder architecture for image compression (discussed in Section 2.1), other compression models use different network architectures. These include compression models using recurrent neural networks [12, 27], generative adversarial networks [17]. Adversarial machine learning (ML) [4] is an active research field where various attack vectors have been explored, including adversarial perturbation attacks [9, 14, 26], data poisoning/backdooring attacks [11, 19], membership inference attacks [23], targeting either confidentiality or integrity of the ML systems. Attacks on availability include [24], which increases the energy-latency of NNs. Our attacks on learned image compression can be considered availability attacks that affect the storage availability. Another work [5] examines the robustness of image compression on its reconstruction; we, instead, investigate bitrate robustness.

7 Conclusion

In this paper, we investigate the bitrate robustness of learned image compression to adversarial images, which significantly consumes more bits to represent the compressed latent. We demonstrate the feasibility by proposing white-box and black-box attacks. We further propose a novel network architecture consisting of a factorized entropy model and attention modules, exhibiting greater performance in both PSNR/bpp ratio and robustness.
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. arXiv preprint arXiv:1802.01436, 2018.

[3] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. arXiv preprint arXiv:2011.03029, 2020.

[4] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 84:317–331, 2018.

[5] T. Chen and Z. Ma. Towards robust neural image compression: Adversarial attack and model finetuning. arXiv preprint arXiv:2112.08691, 2021.

[6] 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, pages 7939–7948, 2020.

[7] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu. Benchmarking adversarial robustness on image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 321–331, 2020.

[8] J. Duda. Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding. arXiv preprint arXiv:1311.2540, 2013.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[10] V. K. Goyal. Theoretical foundations of transform coding. IEEE Signal Processing Magazine, 18(5):9–21, 2001.

[11] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access, 7:47230–47244, 2019.

[12] K. Islam, L. M. Dang, S. Lee, and H. Moon. Image compression with recurrent neural network and generalized divisive normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1875–1879, 2021.

[13] E. Kodak. Kodak lossless true color image suite (photocd pcd0992). URL http://r0k.us/graphics/kodak, 6, 1993.

[14] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In Artificial intelligence safety and security, pages 99–112. Chapman and Hall/CRC, 2018.

[15] J. Lee, S. Cho, and S.-K. Beack. Context-adaptive entropy model for end-to-end optimized image compression. arXiv preprint arXiv:1809.10432, 2018.

[16] K. Liu, H. Yang, Y. Ma, B. Tan, B. Yu, E. F. Young, R. Karri, and S. Garg. Adversarial perturbation attacks on ml-based cad: A case study on cnn-based lithographic hotspot detection. ACM Transactions on Design Automation of Electronic Systems (TODAES), 25(5):1–31, 2020.

[17] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson. High-fidelity generative image compression. Advances in Neural Information Processing Systems, 33:11913–11924, 2020.

[18] D. Minnen, J. Ballé, and G. D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. Advances in neural information processing systems, 31, 2018.

[19] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia. Exploiting machine learning to subvert your spam filter. LEET, 8(1-9):16–17, 2008.

[20] J.-R. Ohm and G. J. Sullivan. Versatile video coding—towards the next generation of video compression. In Picture Coding Symposium, volume 2018, 2018.
[21] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[22] M. Rabbani and R. Joshi. An overview of the jpeg 2000 still image compression standard. *Signal processing: Image communication*, 17(1):3–48, 2002.

[23] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In 2017 IEEE symposium on security and privacy (SP), pages 3–18. IEEE, 2017.

[24] I. Shumailov, Y. Zhao, D. Bates, N. Papernot, R. Mullins, and R. Anderson. Sponge examples: Energy-latency attacks on neural networks. In 2021 IEEE European Symposium on Security and Privacy (EuroS&P), pages 212–231. IEEE, 2021.

[25] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[26] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[27] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5306–5314, 2017.

[28] G. K. Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[29] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019.

[30] D. Zoran, M. Chrzanowski, P.-S. Huang, S. Gowal, A. Mott, and P. Kohli. Towards robust image classification using sequential attention models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9483–9492, 2020.
A Supplementary experimental results

A.1 Rate-distortion performance

In Table 5, we list the rate-distortion performance of pre-trained models factor, hyper0, hyperPri, hyperCm, residOrg, and residAtn (obtained from [3]) and our proposed factorAtn model, each with different bitrate qualities. Each quality corresponds to a unique λ value used in the training loss optimization, as shown in Equation 3 and Equation 5.

Table 5: PSNR (dB) and bitrate (bpp) of learned image compressors with different qualities and their corresponding λ values used in training loss optimization.

| quality = 1 | PSNR | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn | factorAtn |
|-------------|------|--------|--------|----------|---------|----------|----------|-----------|
| (λ = 0.0018) | 26.910 | 27.582 | 27.701 | 28.086 | 28.579 | 28.435 | 26.934 |          |
| Rate        | 0.123 | 0.131 | 0.124 | 0.111 | 0.120 | 0.116 | 0.103 |          |

| quality = 2 | PSNR | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn | factorAtn |
|-------------|------|--------|--------|----------|---------|----------|----------|-----------|
| (λ = 0.0035) | 28.217 | 29.196 | 29.358 | 29.648 | 29.969 | 29.763 | 28.246 |          |
| Rate        | 0.189 | 0.209 | 0.198 | 0.187 | 0.184 | 0.174 | 0.162 |          |

A.2 White-box and black-box attack results

We show additional results on white-box and black-box attacks on different model architectures and bitrate qualities in Table 6, Table 7, Table 8, and Table 9, respectively.

Table 6: Bitrate (bpp) of original and adversarial images and their bpp change (×) in white-box attacks on different model architectures and bitrate qualities.

| quality = 1 | original | attack | change |
|-------------|----------|--------|--------|
| (λ = 0.0018) | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn | factorAtn |
| Rate        | 0.123 | 0.131 | 0.124 | 0.111 | 0.120 | 0.116 | 0.103 |          |
| Rate        | 0.198 | 0.588 | 0.588 | 6.226 | 0.715 | 0.327 | 0.190 |          |

| quality = 2 | original | attack | change |
|-------------|----------|--------|--------|
| (λ = 0.0035) | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn | factorAtn |
| Rate        | 0.189 | 0.209 | 0.198 | 0.187 | 0.184 | 0.174 | 0.162 |          |
| Rate        | 0.440 | 0.478 | 0.461 | 0.432 | 0.417 | 0.427 | 0.397 |          |
Table 7: PSNR (dB) between original images and their original and adversarial reconstruction, respectively, and corresponding PSNR change in white-box attacks on different model architectures and bitrate qualities.

| Quality | PSNR  | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn | factorAtn |
|---------|-------|--------|--------|----------|---------|----------|----------|-----------|
|         | (x, \(\hat{x}\)) |        | (x, \(\hat{x}'\)) |        |         |         |         |           |
| quality = 1 | 26.910 | 27.582 | 27.701 | 28.086 | 28.579 | 28.435 | 26.934 |           |
| change  | -2.0% | 0.9%  | -0.6% | 0.2%  | -1.4% | -1.0% | -2.8% |           |
| quality = 2 | 28.217 | 29.196 | 29.358 | 29.648 | 29.969 | 29.763 | 28.246 |           |
| change  | -3.2% | 0.3%  | -0.8% | 0.6%  | -2.9% | -2.9% | -4.0% |           |
| quality = 3 | 29.617 | 30.973 | 31.130 | 31.362 | 31.344 | 31.317 | 29.590 |           |
| change  | -5.1% | -0.5% | -2.2% | -0.9% | -5.1% | -5.6% | -6.6% |           |
| quality = 4 | 31.277 | 32.839 | 32.950 | 33.086 | 33.389 | 33.365 | 31.125 |           |
| change  | -6.7% | -2.7% | -4.2% | -3.1% | -14.7% | -7.5% | -8.3% |           |
| quality = 5 | 32.956 | 34.526 | 34.970 | 35.093 | 35.117 | 34.949 | 32.768 |           |
| change  | -17.0% | -7.6% | -6.7% | -8.1% | -13.1% | -11.0% | -10.9% |           |
| quality = 6 | 35.380 | 36.744 | 36.911 | 36.988 | 36.707 | 36.623 | 35.051 |           |
| change  | -13.6% | -11.4% | -16.0% | -13.6% | -43.0% | -15.1% | -14.9% |           |

A.3 Reconstruction with large distortion

As observed in Table 2 and Table 4, adversarial images typically introduce negligible distortion in the reconstruction than their originals. However, there is one outlier, as shown in Table 2, that the adversarial images on the residOrg model result in a PSNR drop of 43% in the white-box setting. This kind of large reconstruction distortion, potentially with visible noise, may occur as we did not put constraints on the reconstruction quality of adversarial images in the attack loss function; there is no guarantee that minimum distortion is achieved. We present exemplar adversarial images and their reconstruction, which produce visible distortion in Fig. 5. These images are compressed and reconstructed on the residOrg model with a bitrate quality of 6.

Figure 5: Adversarial images (top row) and their reconstruction with visible distortion (bottom row), compressed and reconstructed with residOrg model (quality 6) in the white-box setting
Table 8: bpp change (×) in black-box attacks on different model architectures and bitrate qualities where adversarial images are generated from substitute DCT-Nets with multiple quantization table Q.

| DCT-Net factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn | factorAtn |
|----------------|--------|----------|---------|----------|----------|----------|
| Q = 10         | 1.157  | 1.319    | 1.347   | 1.382    | 1.414    | 1.415    | 1.193    |
| Q = 30         | 1.081  | 1.213    | 1.232   | 1.256    | 1.303    | 1.296    | 1.095    |
| Q = 50         | 1.054  | 1.157    | 1.170   | 1.187    | 1.238    | 1.222    | 1.064    |
| Q = 70         | 1.038  | 1.119    | 1.126   | 1.139    | 1.190    | 1.167    | 1.045    |
| Q = 90         | 1.018  | 1.065    | 1.066   | 1.072    | 1.101    | 1.082    | 1.021    |
| Q = 10         | 1.228  | 1.394    | 1.422   | 1.451    | 1.480    | 1.481    | 1.279    |
| Q = 30         | 1.148  | 1.344    | 1.366   | 1.397    | 1.423    | 1.413    | 1.170    |
| Q = 50         | 1.103  | 1.281    | 1.299   | 1.330    | 1.352    | 1.335    | 1.117    |
| Q = 70         | 1.073  | 1.220    | 1.238   | 1.268    | 1.284    | 1.262    | 1.083    |
| Q = 90         | 1.035  | 1.123    | 1.133   | 1.149    | 1.152    | 1.132    | 1.039    |
| Q = 10         | 1.299  | 1.426    | 1.454   | 1.484    | 1.501    | 1.498    | 1.355    |
| Q = 30         | 1.248  | 1.444    | 1.466   | 1.495    | 1.497    | 1.494    | 1.272    |
| Q = 50         | 1.187  | 1.414    | 1.430   | 1.458    | 1.442    | 1.433    | 1.201    |
| Q = 70         | 1.134  | 1.366    | 1.376   | 1.405    | 1.368    | 1.354    | 1.144    |
| Q = 90         | 1.065  | 1.230    | 1.229   | 1.250    | 1.200    | 1.186    | 1.070    |
| Q = 10         | 1.347  | 1.458    | 1.478   | 1.519    | 1.539    | 1.534    | 1.423    |
| Q = 30         | 1.352  | 1.529    | 1.545   | 1.582    | 1.614    | 1.611    | 1.400    |
| Q = 50         | 1.311  | 1.545    | 1.549   | 1.579    | 1.610    | 1.607    | 1.335    |
| Q = 70         | 1.242  | 1.529    | 1.520   | 1.544    | 1.580    | 1.571    | 1.256    |
| Q = 90         | 1.120  | 1.392    | 1.373   | 1.391    | 1.447    | 1.421    | 1.130    |
| Q = 10         | 1.376  | 1.473    | 1.547   | 1.575    | 1.581    | 1.578    | 1.463    |
| Q = 30         | 1.422  | 1.560    | 1.654   | 1.680    | 1.692    | 1.684    | 1.496    |
| Q = 50         | 1.420  | 1.594    | 1.701   | 1.717    | 1.726    | 1.714    | 1.471    |
| Q = 70         | 1.377  | 1.612    | 1.743   | 1.746    | 1.730    | 1.715    | 1.403    |
| Q = 90         | 1.216  | 1.544    | 1.744   | 1.740    | 1.628    | 1.612    | 1.221    |
| Q = 10         | 1.419  | 1.556    | 1.605   | 1.625    | 1.642    | 1.634    | 1.521    |
| Q = 30         | 1.500  | 1.672    | 1.731   | 1.752    | 1.755    | 1.750    | 1.601    |
| Q = 50         | 1.533  | 1.733    | 1.793   | 1.810    | 1.793    | 1.788    | 1.620    |
| Q = 70         | 1.548  | 1.796    | 1.851   | 1.863    | 1.806    | 1.796    | 1.616    |
| Q = 90         | 1.471  | 1.901    | 1.931   | 1.947    | 1.776    | 1.749    | 1.488    |

A.4 Robustness to random noise

Learned image compression models show more robustness to random noise than adversarial images. As shown in Table 10, there is little bpp change after adding Gaussian noise (mean = 0, variance = 1, under the same perturbation constraint as adversarial images) to the input images, verified with different model architectures and bitrate qualities.

A.5 Attack on JPEG compression

In addition to white-box and black-box attacks on learned image compression models, we also attack JPEG compression with different bitrate qualities. Using the same black-box attack method proposed in Section 3, we examine the bpp change of JPEG compression with adversarial images generated by various DCT-Nets, as shown in Table 11. JPEG compression shows more robustness than learned image compression models with similar bitrate qualities.

B Additional experimental setup

B.1 Experimental platform

We perform NN training/inference on GPU workstation with Intel CPU i9-10980XE (36 cores, 3.00GHz) and Nvidia Geforce RTX3090 GPUs. We implement our experiments using PyTorch 1.10.1, CUDA 11.4, and Python 3.8.12 on Ubuntu 20.04.
Table 9: PSNR (dB) between original images and their original and adversarial reconstruction, respectively, and corresponding PSNR change in black-box attacks on different model architectures and bitrate qualities.

| Quality | PSNR | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn | factorAtn |
|---------|------|--------|--------|----------|---------|----------|----------|-----------|
| 1       |      | 26.910 | 27.582 | 27.701   | 28.086  | 28.579   | 28.435   | 26.934    |
|         | change | 0.7%   | 1.7%   | 1.7%     | 1.2%    | 1.7%     | 1.5%     | 0.5%      |
| 2       |      | 28.217 | 29.196 | 29.358   | 29.648  | 29.969   | 29.763   | 28.246    |
|         | change | 0.2%   | 1.2%   | 1.6%     | 1.5%    | 0.6%     | 0.9%     | 0.1%      |
| 3       |      | 29.617 | 30.973 | 31.130   | 31.362  | 31.344   | 31.317   | 29.590    |
|         | change | -1.0%  | 0.8%   | 1.1%     | 0.8%    | -1.7%    | -1.6%    | -1.1%     |
| 4       |      | 31.277 | 32.839 | 32.950   | 33.086  | 33.389   | 33.365   | 31.125    |
|         | change | -1.8%  | -1.3%  | -1.1%    | -2.8%   | -3.5%    | -3.8%    | -2.9%     |
| 5       |      | 33.657 | 34.526 | 34.970   | 35.093  | 35.117   | 34.949   | 32.768    |
|         | change | -4.9%  | -5.3%  | -2.6%    | -5.8%   | -6.3%    | -5.9%    | -4.5%     |
| 6       |      | 35.870 | 36.744 | 36.911   | 36.988  | 36.707   | 36.623   | 35.051    |
|         | change | -6.8%  | -10.1% | -9.6%    | -9.8%   | -10.9%   | -10.6%   | -7.4%     |

Table 10: bpp change (×) after adding Gaussian noise (mean = 0, variance = 1, constrained by the same perturbation allowance as adversarial images) to input images on different model architectures and bitrate qualities.

| Quality | factor | hyper0 | hyperPri | hyperCm | residOrg | residAtn | factorAtn |
|---------|--------|--------|----------|---------|----------|----------|-----------|
| 1       | 1.001  | 1.015  | 1.013    | 1.010   | 1.012    | 1.005    | 0.999     |
| 2       | 1.005  | 1.030  | 1.031    | 1.028   | 1.025    | 1.018    | 1.004     |
| 3       | 1.012  | 1.052  | 1.047    | 1.046   | 1.037    | 1.030    | 1.011     |
| 4       | 1.027  | 1.090  | 1.083    | 1.091   | 1.083    | 1.081    | 1.026     |
| 5       | 1.055  | 1.168  | 1.190    | 1.192   | 1.139    | 1.139    | 1.050     |
| 6       | 1.141  | 1.379  | 1.367    | 1.363   | 1.271    | 1.246    | 1.118     |

B.2 Network architectures

Network architectures of pre-trained image compression models factor, hyper0, hyperPri, hyperCm, residOrg, and residAtn can be found in [3]. We show the architecture of our proposed factorAtn model in Table 12 which essentially is a modified factor model with attention modules incorporated. Here attention (N) in Table 12 denotes that N 1 × 1 convolutional filters are used in the attention module architecture, as shown in Fig. 6. All image compression models in this paper use range asymmetric numerical coding [8].

B.3 Hyper-parameters

**Attack** We normalize the pixel intensities of input images with dimensions 512 × 768 × 3 between 0 and 1. In the white-box attacks (Algorithm 1), we use maximum perturbation ϵ = 30 as measured in the L2 norm, corresponding to 7/255 per pixel change averaged over all the pixels. We use step size δ = 0.004, and maximum iteration T = 600. In practice, we terminate our attack early when the attack loss function L_{atk} stops increasing after 20 consecutive iterations, even though the total running iterations have not reached the maximum allowed iteration T.

3https://github.com/rygorous/ryg_rans
Table 11: bpp change (×) in black-box attacks on JPEG compression with different bitrate qualities where adversarial images are generated from substitute DCT-Nets with multiple quantization table Q.

| DCT-Net | JPEG Q = 10 | JPEG Q = 20 | JPEG Q = 50 | JPEG Q = 70 | JPEG Q = 90 |
|---------|-------------|-------------|-------------|-------------|-------------|
| Q = 10  | 1.267       | 1.393       | 1.401       | 1.385       | 1.367       |
| Q = 30  | 1.092       | 1.445       | 1.453       | 1.421       | 1.437       |
| Q = 50  | 1.056       | 1.483       | 1.468       | 1.449       | 1.463       |
| Q = 70  | 1.039       | 1.256       | 1.507       | 1.448       | 1.487       |
| Q = 90  | 1.018       | 1.091       | 1.195       | 1.308       | 1.527       |

Table 12: Network architecture of factorAttn model

| Layer/Block | Analysis Transform | Synthesis Transform |
|-------------|--------------------|---------------------|
| 1           | $5 \times 5 \times 128$ Conv, GDN | Attention (192) |
| 2           | $5 \times 5 \times 128$ Conv, GDN | $5 \times 5 \times 128$ Deconv, IGDN |
| 3           | Attention (128) | $5 \times 5 \times 128$ Deconv |
| 4           | $5 \times 5 \times 128$ Conv, GDN | Attention (128), IGDN |
| 5           | $5 \times 5 \times 192$ Conv | $5 \times 5 \times 128$ Deconv, IGDN |
| 6           | Attention (192) | $5 \times 5 \times 3$ Deconv |

Figure 6: attention network architecture

**Training** We used the same training hyper-parameters as in [3] to train our DCT-Nets and factorAttn model. Training dataset is Vimeo90K [29], and we train approximately 80 ∼ 100 epochs using Adam optimizer.

**B.4 Source Code**

Source code will be made available publicly upon publication.

**C Societal and ethical impacts**

Our work investigates adversarial attacks on learned image compression models (albeit in a “white-hat” manner), which allows an adversary to disrupt image compression systems by crafting adversarial images that significantly consume more bits than expected after compression. Given increasing security and robustness concerns with machine learning techniques, we believe that our work provides timely insights on the implications of such attacks and will hopefully encourage more work on the robustness of learned image compression. However, given that such learned image compression models have not become industry standards and are not yet in widespread general use, we expect the negative societal impact to be minimal, with no ethical impact involved.