Resilience of Entropy Model in Distributed Neural Networks

Milin Zhang, Mohammad Abdi,
Shahriar Rifat, and Francesco Restuccia

Institute for the Wireless Internet of Things
Northeastern University, Boston MA 02115, USA
{zhang.mil,abdi.mo,rifat.s,f.restuccia}@northeastern.com

Abstract. Distributed deep neural networks (DNNs) have emerged as a key technique to reduce communication overhead without sacrificing performance in edge computing systems. Recently, entropy coding has been introduced to further reduce the communication overhead. The key idea is to train the distributed DNN jointly with an entropy model, which is used as side information during inference time to adaptively encode latent representations into bit streams with variable length. To the best of our knowledge, the resilience of entropy models is yet to be investigated. As such, in this paper we formulate and investigate the resilience of entropy models to intentional interference (*e.g.*, adversarial attacks) and unintentional interference (*e.g.*, weather changes and motion blur). Through an extensive experimental campaign with 3 different DNN architectures, 2 entropy models and 4 rate-distortion trade-off factors, we demonstrate that the entropy attacks can increase the communication overhead by up to 95%. By separating compression features in frequency and spatial domain, we propose a new defense mechanism that can reduce the transmission overhead of the attacked input by about 9% compared to unperturbed data, with only about 2% accuracy loss. Importantly, the proposed defense mechanism is a standalone approach which can be applied in conjunction with approaches such as adversarial training to further improve robustness. Code is available at `https://github.com/Restuccia-Group/EntropyR`.

Keywords: Trustworthy Machine Learning · Data Compression

1 Introduction

Distributed deep neural network (DNN) were recently introduced to divide the computation of DNNs across various devices based on their available computation and communication resources. They have been shown to be extremely effective to implement deep learning applications on resource-constrained mobile devices [18,39,54]. As depicted in Fig. 1, the common strategy is to divide a large DNN into a small *head* network deployed on the mobile device to extract and compress features, and a *tail* network on the server to perform the task inference. Compared to conventional lightweight DNNs specifically designed for

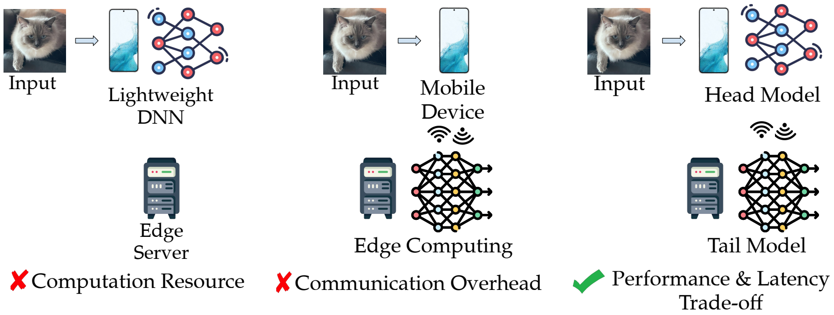


Fig. 1: The key concept of distributed DNN. Left: Lightweight DNNs suffer from performance loss due to the limited computational resources; Middle: Edge computing often results in intolerable latency due to communication overhead; Right: Distributed DNNs deploy a small *head* model on the resource-limited device to achieve task-oriented compression and a large *tail* model on the server to decode compressed features and execute the rest of model.

mobile devices [28,29,31,52,57], distributed DNNs can leverage the computation power of the edge/cloud and hence attain better performance. In addition, distributed DNNs leverage compression techniques to reduce the data size of intermediate representations. By only transmitting compact representations, distributed DNNs can reduce the transmission overhead significantly compared to traditional edge computing [34,48,60].

Interestingly, recent work has proposed to further minimize the data size of transmitted latent representations with entropy coding [42]. The key idea is to train the distributed DNN jointly with an auxiliary entropy model, which estimates the distribution of latent representations. The output of the entropy model is then used as side information during inference time to facilitate the adaptive encoding of latent representations into bit streams with variable lengths. The proposed approach achieves better compression rate compared to quantization-based [38] and codec-based [2] approaches by minimizing the entropy of latent representations. *Despite its excellent compression performance, the resilience of entropy coding to intentional interference (*e.g.*, specifically crafted adversarial attacks) and unintentional interference (*e.g.*, sudden events such as weather changes and motion blur) remains unexplored.* We remark that in entropy-coding-based distributed DNNs, the bit rate of encoded representations relies completely on its estimated entropy. However, it is well known that DNNs trained on standard datasets are vulnerable to distribution shifts [24,25] and adversarial actions [20,56]. As the entropy model is learned without considering any corruptions and adversarial perturbations, *a small change in the input space might lead to a large estimate of entropy, and hence result in a bit rate that can exceed the transmission bandwidth.* As depicted in Tab. 3 in our experiments, the transmission overhead can be increased by about 2x in the worst case.

In this work, we thoroughly assess for the first time the resilience of entropy models to intentional and unintentional interference as applied to distributed DNNs. Intriguingly, our findings unveil that the auxiliary entropy model learns a *different set of input-related features* than what learned by the backbone DNN. Ultimately, this enables us to design effective defense strategy to maintain the bit rate with an unnoticeable loss in the task performance. Our approach successfully reduces the communication overhead of perturbed data by about 9% compared to unperturbed inputs, with only about 2% accuracy loss.

Summary of Novel Contributions

(1) *We investigate for the first time the effect of intentional and unintentional interference to entropy coding in distributed DNNs.* We show that interference increases the end-to-end latency of distributed DNNs and poses a threat to other users by saturating the transmission bandwidth. Through comprehensive experiments involving 3 distinct DNN architectures [23,47], 2 entropy models [5,43], and 4 rate-distortion trade-off factors, we illustrate that attacks to entropy coding may increase the transmitted data size by up to about 95%;

(2) We design two visualization approaches to interpret the compression learned by entropy models. By disentangling features relevant to compression and classification across both frequency and spatial domains, we reveal that the entropy model learns specific features *that are distinct* from those beneficial for the end task. This finding enables us to devise an effective defense strategy that safeguards the vulnerable compression features while minimally impacting task performance;

(3) We propose a defense approach to attacks targeting entropy coding based on our findings discussed above. The proposed method effectively reduces the transmission overhead of attacked inputs by about 9% in comparison to the unperturbed case, with only about 2% decrease in accuracy. We remark that our approach is general in nature and can be combined with approaches such as adversarial training to further improve the resilience of entropy models.

The paper is organized as follows. Sec. 2 provides background of entropy coding in distributed DNNs. In Sec. 3, we formally describe the threat model for entropy coding in distributed DNNs. Sec. 4 presents benchmarking results on both intentional and unintentional interference. Sec. 5 details the proposed defense approach and comprehensively evaluates it against adaptive attacks. Related work is summarized in Sec. 6. Finally, we draw conclusions and discuss future work in Sec. 7.

2 Entropy Coding in Distributed DNNs

Entropy coding techniques such as arithmetic coding [50] and asymmetric numeral systems [17] can encode a message with the optimal coding rate by leveraging the probabilistic information of the message. Based on the source coding theorem [53], the entropy of a message z with distribution $P_Z(z)$ defines the lower bound of expected coding rate $R_c(z)$ without any loss of information, i.e.,

$$\mathbb{E}\{R_c(z)\} \geq H_{P_Z}(z) = \mathbb{E}\{-\log_2 P_Z(z)\} \tag{1}$$

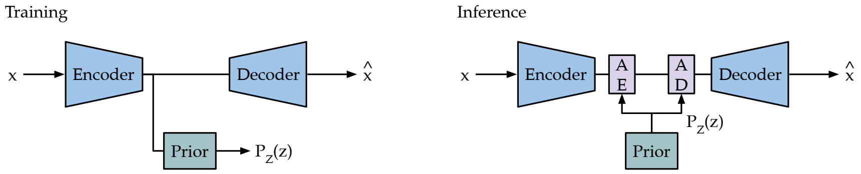


Fig. 2: A general framework for entropy coding in neural data compression. It consists of an encoder-decoder structure to extract a compact latent representation z and a prior model to estimate the distribution $P_Z(z)$. Left: During training, the prior model is jointly optimized with the encoder-decoder backbone. Right: During inference, the learned prior $P_Z(z)$ is used as side information for arithmetic encoding/decoding.

where $H_{P_Z}(z)$ is the entropy of z. In recent data compression literature [1,5,6,14,43,44], the entropy coding is integrated with lossy DNN-based compression to attain an effective compression rate with less information loss compared to traditional approaches. As depicted in Fig. 2, the common strategy is to jointly train an encoder-decoder backbone for extracting compact latent representations z with an auxiliary prior model $P_Z(z)$. During training, a random noise source is introduced to simulate the effect of quantization on z, allowing the backpropagation of gradients. In the inference phase, z is initially quantized to integers for entropy coding. Next, the output of the prior model is used as side information to encode the quantized z into bit streams adaptively with variable length.

DNN-based data compression can achieve the near-optimal compression rate by minimizing the entropy of latent representations during training. Specifically, the objective is to minimize the rate-distortion function

$$\mathcal{L}(x, (\hat{x}, z)) = \underbrace{||x - \hat{x}||_2^2}_{\text{distortion}} - \beta \cdot \underbrace{\log_2 P_Z(z)}_{\text{rate}}, \tag{2}$$

where $\hat{x}$ is the reconstructed data, $||x - \hat{x}||_2^2$ is the loss function of the encoder-decoder backbone and $-\log_2 P_Z(z)$ is the loss function of the prior model. Constant β denotes the trade-off between the rate and distortion.

Recent work [42] has incorporated entropy coding into distributed DNNs to increase communication efficiency between mobile and edge devices. The proposed method, referred to as the *entropic student*, aims at achieving a balance between the coding rate and task-specific performance metrics (*e.g.*, cross entropy in classification tasks). One distinction between the *entropic student* and other entropy-coding-based feature compression work [55,59] lies in the significantly smaller encoder size of the former due to constraints on computational resources in mobile devices. As a result, the *entropic student* splits the DNN at an early layer and use knowledge distillation [26] for training the *head* to preserve accuracy. **However, existing work does not address the issue of resilience of entropy models, which is the key goal of this paper**. A relevant work [4] investigated the robustness of density estimation aiming to maximize $P_Z(z)$ while we are interested in perturbations that minimize $P_Z(z)$.

3 Modeling Interference to Entropy Coding

While the resilience of DNNs to distribution shifts [7,19,24,25,58] and adversarial actions [3,11,12,30,35,56] has been extensively investigated, we investigate types of interference that can alter the length of the encoded bit stream, which ultimately leads to increased bandwidth utilization. In this section, we formulate the resilience of compression in entropy-coding-driven distributed DNNs.

Threat Model for Accuracy. Let $D = \{x, y\}$ denotes a dataset for classification problem where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are specific input and corresponding label samples respectively, while $\mathcal{X}$, $\mathcal{Y}$ are input and output space, respectively. There exists a prior yet unknown distribution $P(y|x)$ such that

$$P(y|x) \geq P(\hat{y}|x) \qquad \forall \hat{y} \in \mathcal{Y} \tag{3}$$

where $\hat{y}$ denotes incorrect labels. The DNN classifier, is trained to approximate the distribution $\tilde{P}(y|x) \approx P(y|x)$ with empirical risk minimization approaches based on an assumption that in-sample data and out-of-sample data shared a similar distribution $P(y|x)$. However, the unintentional interference denoted by δ, which can be modeled as an additive noise to x, creates a new dataset $D' = \{x + \delta, y\}$. It introduces a covariate shift that $P(x + \delta) \neq P(x)$ but $P(y|x + \delta) = P(y|x)$, resulting in classification errors such that

$$\tilde{P}(\hat{y}|x + \delta) \geq \tilde{P}(y|x + \delta) \qquad \exists \hat{y} \in \mathcal{Y} \tag{4}$$

In the intentional interference scenario, algorithms attempt to find the minimum covariate shift that leads to Eq. (4). In this setting, a small perturbation in l_p distance $||\delta||_p \leq \epsilon$ is intentionally crafted to mislead the DNN classifier where ϵ denotes the constraint. Attackers formulate Eq. (4) as an optimization problem over the input space $\mathcal{X}$, and hence can apply the gradient information of the loss function w.r.t. x to find the δ. Projected Gradient Descent (PGD) [35] provided a unified view on iterative gradient attacks under l_p constraint:

$$x^{i+1} = x^i + \alpha \cdot \epsilon \cdot \frac{\nabla_x \mathcal{L}}{||\nabla_x \mathcal{L}||_p} \tag{5}$$

where x^i is the adversarial sample at i step, α is the learning rate, and $\frac{\nabla_x \mathcal{L}}{||\nabla_x \mathcal{L}||_p}$ is the normalized gradient of a loss function $\mathcal{L}$ (*e.g.*, a sign function in l_∞ space). $\mathcal{L}$ can be cross entropy or other more advanced loss functions [11].

Threat Model for Data Rate. Let $H(x) = z$ denotes the *head* network where $z \in \mathcal{Z}$ is the output of the *head* and $\mathcal{Z}$ is the latent space. The goal is to achieve a minimum compression rate $-\log_2 P_Z(z)$. As z is dependent on x, the distribution shift introduced by unintentional interference can result in a difference in coding rate:

$$-\log_2 P_Z(H(x + \delta)) \neq -\log_2 P_Z(H(x)) \tag{6}$$

Table 1: DNNs, entropy models and datasets utilized in experiments.

DNN Architectures	
ResNet-50 [23]	A standard baseline in distributed DNNs [38–40,42]
ResNet-101 [23]	A deeper model in the ResNet family
RegNetY-6.4GF [47]	An advanced design with better performance
Entropy Models	
FP [5]	A fully factorized prior model
MSHP [43]	An effective learnable entropy model
Datasets	
ImageNet [16]	The standard ImageNet validation dataset
ImageNet-C [25]	A synthetic dataset with 15 corruptions and 5 severities
Random Noise	Noisy images as a baseline for adversarial robustness
PGD-Acc [35]	PGD targeting classification
PGD-E	PGD targeting entropy

Note that the unintentional distribution shift of x does not necessarily result in a larger bit rate. However, in the adversarial setting, attackers intend to *maximize* the entropy of z as follows:

$$\begin{aligned} \min_{\delta} \quad & \log_2 P_Z(H(x+\delta)) \\ s.t. \quad & ||\delta||_p \le \epsilon \end{aligned} \tag{7}$$

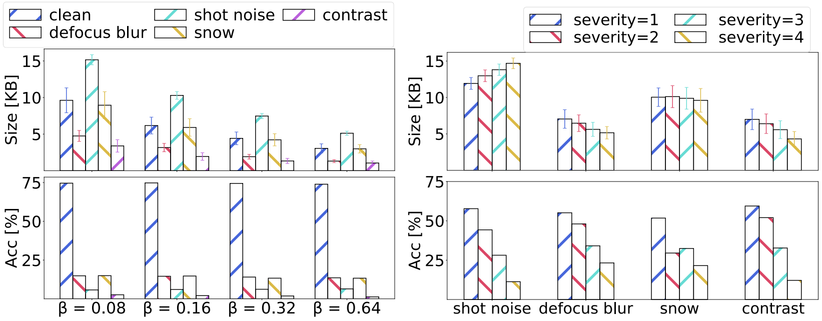
Thus, to correctly assess the robustness of entropy models, we use the same PGD algorithm in Eq. (5) yet with the entropy as its loss function $\mathcal{L} = -\log_2 P_Z(H(x))$.

4 Understanding the Resilience of Entropy Model

4.1 Experimental Setup

In this section, we outline the experimental setup for evaluating the resilience of entropy models. We first specify the DNNs and entropy models under consideration, followed by a description of the metrics and datasets used in our experiments. A summary of our experimental setup is provided in Tab. 1.

Ablation of DNNs and Entropy Models. We consider three distinct DNNs: ResNet-50 [23], ResNet-101 [23] and RegNetY-6.4GF [47]. ResNet-50 serves as a common baseline in distributed DNN literature [38–40,42] while ResNet-101 is a deeper variant with the same design as former. RegNetY-6.4GF has an advanced design with better classification performance. The early layers are replaced by a specific tailored *head* architecture as proposed in [42] and incorporated with different entropy models. For the sake of brevity, we consider ResNet-50 as the default DNN architecture without additional specification. We employ two entropy



(a) resilience w.r.t. β **(b)** resilience w.r.t. severities

Fig. 3: Resilience of accuracy and data rate to 4 common corruptions.

models: Factorized Prior (FP) [5] and Mean Scale Hyper Prior (MSHP) [43]. FP is the earliest design of entropy model and a basic component in many other advanced designs, while MSHP provides more powerful compression performance by injecting a hierarchical learnable block before FP to extract the entropy information. For conciseness, we consider MSHP as the default entropy model. In addition, since models are trained with different rate-distortion trade-off β, we also investigate the compression robustness w.r.t. different β.

Datasets and Metrics. Conversely from existing work that measures the compression performance with bits per pixel (BPP) [5,6,14,43,44], we evaluate the entropy model with the data size of the whole bit stream after encoding as in [37]. Compared to BPP, data size quantifies more directly the networking traffic between the *head* and *tail.* To assess the resilience to unintentional interference, we use the ImageNet-C dataset proposed by [25] comprising 15 common corruptions classified into 4 categories (noise, blur, digital, and weather) with 5 different severities. We also report the data size and accuracy on the clean ImageNet [16] validation set as a baseline. To assess the resilience to intentional interference, we implement the PGD targeting both the classification and compression performance in l_∞ space. For clarity, we denote the conventional PGD as PGD-Acc and the PGD targeting entropy models as PGD-E. Meanwhile, random noise with the same perturbation level is added to the clean ImageNet as a baseline comparison for adversarial robustness.

4.2 Resilience to Unintentional Interference

Resilience w.r.t. $\boldsymbol{\beta}$. We select one corruption as a representative from each category in ImageNet-C [25] to investigate the resilience as a function of rate-distortion trade-off β. Fig. 3a shows the compression and classification performance on *defocus blur*, *shot noise*, *snow* and *contrast* dataset, each with the severity of 5. Compared to the clean ImageNet, all corruptions decrease the accuracy significantly. However, not all corruptions increase the data size. Only

Table 2: Resilience w.r.t. prior models. The average data size of FP and MSHP for clean ImageNet are 11.65 ± 1.02 and 9.62 ± 1.70 KBytes, respectively. Red for larger data size and blue for lower compared to the baseline.

Prior	Gaussian Noise		Motion Blur		Impulse Noise		Glass Blur	
Model	Size[KB]	Acc[%]	Size[KB]	Acc[%]	Size[KB]	Acc[%]	Size[KB]	Acc[%]
FP	13.19±0.40	59.51	10.51±0.70	62.22	13.36±0.34	50.93	10.43±0.66	53.63
MSHP	11.96±0.47	59.36	7.88±1.46	61.58	12.52±0.35	51.12	7.78±1.39	53.67

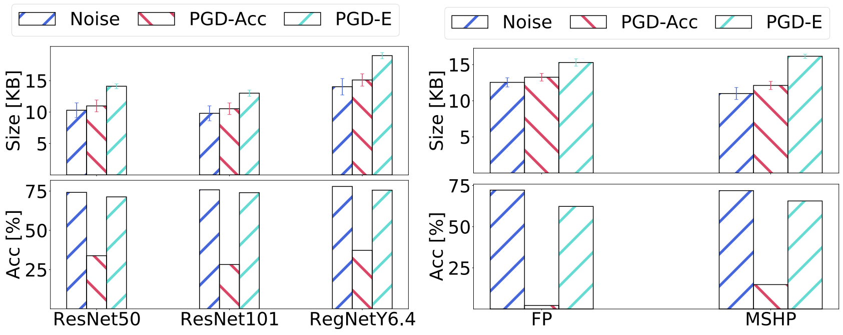
shot noise increases the data size by 65.31% on average. In contrast, *Defocus blur* and *contrast* decrease the data size by 53.31% and 67.45% on average, respectively. *Snow* has little affect to compression, resulting in only 4.42% decrease of data size. This indicates that *the entropy model learns a different set of features than the classification model.* Since it shows the same trend for different rate-distortion trade-off factors, we only consider $\beta = 0.08$ in next experiments.

Resilience w.r.t. Severities. Next, we explore resilience w.r.t. severity levels using the same corruptions mentioned earlier. As illustrated in Fig. 3b, the data size of *shot noise* rises with increasing severity, whereas *defocus blur* and *contrast* consistently decrease the data size. *Snow* shows minimal variation across different severity levels. This can be attributed to *defocus blur* and *contrast* removing high-frequency components in images, while *shot noise* introduces new patterns in high-frequency space. *Snow*, on the other hand, lacks a specific pattern in the frequency domain. Thus, *the compression is particularly sensitive to high-frequency information in the input space.*

Resilience w.r.t. Prior Models. Tab. 2 shows the impact of additional 4 corruptions at severity level 1 on both FP and MSHP model. To further validate the previous observation, we choose 2 noise corruptions (*Gaussian noise* and *impulse noise*), which introduce high-frequency noise to images, and two blur corruptions (*motion blur* and *glass blur*), which eliminate high-frequency components from images. As shown in Tab. 2, *Gaussian noise* increases 13.22% and 24.32% of data size for FP and MSHP, respectively. *Impulse noise* increases 14.42% and 30.15% of data size for FP and MSHP, respectively. On the other hand, *motion blur* reduces 9.79% and 18.09% of data size for FP and MSHP, respectively. *Glass blur* reduces 10.47% and 19.13% of data size for FP and MSHP, respectively. Thus, the previous finding that compressive features reside in the high-frequency space remains consistent for different entropy models.

4.3 Resilience to Intentional Interference

Resilience w.r.t. DNNs Fig. 4a shows the resilience of different DNN architectures to accuracy and entropy attacks with the l_∞ constraint $\epsilon = 4/255$. As evident in the figure, the attacks show similar patterns regardless of their



(a) resilience w.r.t. DNN architectures **(b)** resilience w.r.t. prior models

Fig. 4: Resilience of accuracy and data rate to intentional interference

victim models. While PGD-Acc reduces 42.88% accuracy on average, it only increases by 7.28% of data size on average compared to random noise. On the other hand, PGD-E decreases 2.34% accuracy on average compared to random noise but increases the bit rate 1.35x times on average. This indicates that *PGD-E and PGD-Acc are targeting separate features in the input space which has minor impact on each other.*

Resilience w.r.t. Prior Models. Fig. 4b illustrates the effect of adversarial attacks with $\epsilon = 8/255$ on victims using different prior models. As shown in the figure, both entropy models show considerable vulnerability to PGD-E compared to PGD-Acc. The data size are increased by 21.93% and 46.82% for FP and MSHP respectively while the performance loss are 9.91% and 6.62% respectively. On the other hand, PGD-Acc successfully reduces 72.10% and 57.10% classification performance of FP and MSHP compared to random noise respectively. However, it only increases 5.66% and 10.19% bit rate for FP and MSHP respectively. Hence, attackers targeting compression tend to perturb a different set of features than attackers targeting discriminative tasks regardless of the entropy models. In addition, while the FP has a better robustness in compression, it has significantly worse robustness in accuracy.

5 Proposed Defense Mechanism

5.1 Disentangle Compression Features in Frequency Domain

As shown in Sec. 4.2, interference that introduce high-frequency noise (*e.g.*, *shot noise*) can effectively increase the data size while interference that remove high-frequency information (*e.g.*, *defocus blur*) reduce the data size. Intuitively this is a consequence of the small *head* in distributed DNNs as semantic information is usually captured in deeper layers while early layers tend to capture low level information [8]. To validate our intuition, we introduce total variation, defined

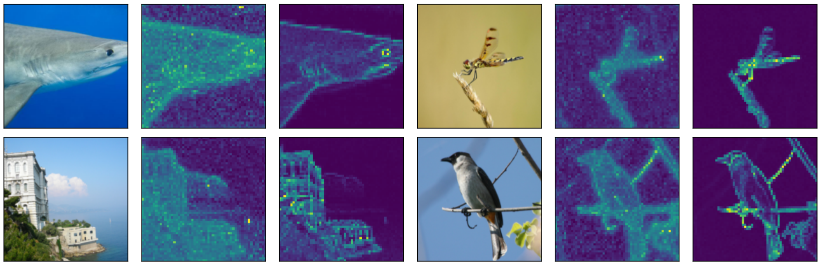


Fig. 5: Entropy models are sensitive to high-frequency features. From left to right: images from ImageNet validation set, bit rate maps and total variation maps.

as the integral of image gradient magnitude, which is first proposed for image denoising in [51]. The anitrosopic total variation of a 2-D image is defined as

$$TV(x) = \sum_{i,j} |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}| \tag{8}$$

where x denotes an image sample, i, j denotes the pixel position in width and height, respectively. The image gradient magnitude $|x_{i+1,j} - x_{i,j}|$ and $|x_{i,j+1} - x_{i,j}|$ describe the sharpness of pixel changes, where larger magnitude indicates high-frequency information such as edges and textures in images.

We compare the total variation map of an image with its bit rate map generated by the entropy model. To plot the total variation map, we first slice an image into many small patches and then compute the total variation for each patch. Meanwhile, the bit rate map is the estimated entropy of the latent representation $-\log_2 P_Z(z)$. As shown in Fig. 5, there exists a significant correlation between the bit rate map and the total variation map, which indicates that the entropy model is sensitive to high-frequency features in the input space.

5.2 Disentangle Compression Features in Spatial Space

The work in [42] has shown that entropy model in distributed DNNs tends to assign a larger number of bits to task-oriented information, such as objects, while compressing irrelevant information, like the background. Consequently, it becomes intuitively easier for an adversary to increase the bit rate in the background area to attack the entropy model.

To interpret the action conducted by attackers, we generate the bit rate comparison map between adversarial samples generated by PGD-E and PGD-Acc. This comparison map represents the estimated entropy of adversarial samples generated by PGD-E subtracted from the bit rate of PGD-Acc-perturbed samples. In this context, the red area indicates that PGD-E tends to allocate more bits compared to PGD-Acc, whereas the blue region suggests that PGD-E tends to allocate fewer bits. As depicted in Fig. 6, PGD-E puts more efforts to increase the bit rate in background region, whereas PGD-Acc focuses on altering

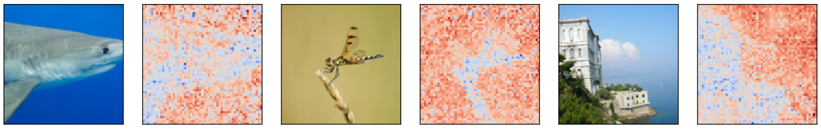


Fig. 6: Entropy models are more vulnerable to perturbation in background region. From left to right: images from ImageNet validation set, bit rate comparison between PGD-E and PGD-Acc (PGD-E allocate less bit in blue area and more bit in red area).

task-oriented information. This is in line with the experiments in Sec. 4.3 that PGD-E and PGD-Acc target different features in the input space. On one hand, increasing the bit rate in the background region will affect little accuracy as they are irrelevant information for the discriminative task. On the other hand, modifying object information can significantly degrade the performance, yet it has less impact on compression.

5.3 Object-Aware Total Variation Denoising

Given the observation in Sec. 5.1, we propose a denoising technique based on total variation to remove high-frequency noise in adversarial images where we solve the following optimization problem:

$$\min_{x} \frac{1}{2}||x - x'||_2^2 + \lambda \cdot TV(x), \tag{9}$$

where x' is the distorted image, x is the denoised image, $TV(x)$ is the total variation of x and λ is a regularization factor to control the degree of smoothing. Eq. (9) means the image x after denoising should keep most of the information in x' ($i.e.$, minimize $||x-x'||_2^2$) but discard the high-frequency information ($i.e.$, minimize $TV(x)$). Optimizing Eq. (9) is not trivial as total variation is non-differentiable. As a result, we use a sub-gradient descent:

$$x^{i+1} = x^i - \alpha \cdot ((x^i - x') + \lambda \cdot g(x^i)) \tag{10}$$

where x^i is the denoised image at the step i, α is the step size and $g(x^i)$ is the image gradient of x^i [9].

Simply applying the total variation denoising will also decrease the task-oriented performance as it also removes useful information in high-frequency domain. Since adversarial attacks targeting entropy models attempt to increase bit rate in background region as demonstrated in Sec. 5.2, we add a mask to Eq. (10) to force the denoising algorithm to only remove high-frequency information in non-object space, that is,

$$x^{i+1} = x^i - \alpha \cdot m \cdot ((x^i - x') + \lambda \cdot g(x^i)) \tag{11}$$

where m is the mask to control the denoising level. In practice, we interpolate the output of entropy model $P_Z(z)$ as the soft mask. This is because the higher bit rate $-\log_2 P_Z(z)$ corresponding to object area has a smaller value of $P_Z(z)$, making it naturally a soft mask to avoid smoothing the object region.

Table 3: Compression and classification performance before and after defense. The average data size for clean ImageNet is 9.62 ± 1.70 KBytes.

	$\epsilon = 2/255$		$\epsilon = 4/255$		$\epsilon = 8/255$		$\epsilon = 16/255$	
	Size[KB]	Acc[%]	Size[KB]	Acc[%]	Size[KB]	Acc[%]	Size[KB]	Acc[%]
before	12.35±0.63	**73.26**	14.11±0.38	**71.33**	16.15±0.29	65.62	18.79±0.37	52.13
after	**8.76±1.38**	70.82	**9.55±1.07**	70.15	**11.02±0.65**	**67.22**	**13.44±0.48**	**59.87**

Experimental Results. We investigate the proposed defense method as a function of perturbation budget ϵ. As shown in Tab. 3, attackers targeting entropy significantly increase the data size, by up to 95.32% when $\epsilon = 16/255$ compared to the clean images (9.62 KBytes). On the contrary, our denoising approach significantly reduce the data size for all ϵ. For $\epsilon = 2/255, 4/255$, the average data size is even 8.94% and 0.73% smaller than the clean images with only 2.44% and 1.18% accuracy loss, respectively. For $\epsilon = 8/255, 16/255$, the data size is reduced by 31.80% and 28.47% respectively, while the accuracy improved slightly after denoising (1.60% for $\epsilon = 8/255$ and 7.74% for $\epsilon = 16/255$).

We remark that the proposed defense is a standalone approach which can be incorporated into other approaches such as adversarial training to further improve the resilience. The accuracy loss incurred in small perturbation scenarios ($\epsilon = 2/255, 4/255$) can be mitigated by training with augmented data.

5.4 Adaptive Attacks

To thoroughly evaluate the effectiveness of a defense strategy, [10] proposed to design adaptive attacks that counter the defense mechanism with perfect knowledge of both DNN and defense in the white-box setting. To this end, we assess our defense with two adaptive attacks.

Low Frequency Attack. As the total variation denoising is proposed to remove high-frequency components, the first adaptive attack that we considered is to only add perturbations in low frequency space. Following the methodology outlined in [21], we first convert the gradient to frequency space with discrete cosine transform and mask out the high-frequency components. The gradient is then transformed back and Eq. (5) is applied to create the adversarial samples.

Regional Attack. Given that the defense exclusively denoises the background region, the second adaptive attack we considered is to force the adversary put greater effort in increasing bit rate in the object region. Similar to Eq. (11), a mask is multiplied to the loss function before backpropagation. In practice, we choose $1 - P_Z(z)$ as the soft mask where $P_Z(z)$ is the probability of z ranging between 0 and 1. A larger $P_Z(z)$ indicates a smaller bit rate $-\log_2 P_Z(z)$ correponding to the background region. Therefore, $1-P_Z(z)$ encourages the adversary to take more attention to object region.

Table 4: Resilience to adaptive attacks with $\epsilon = 4/255$.

Adaptive	Before Denoising		After Denoising	
Attacks	Size[KB]	Acc[%]	Size[KB]	Acc[%]
Frequency	14.11±0.38	71.38	9.56±1.07	70.17
Regional	14.05±0.41	71.63	9.50±1.10	70.22

Resilience to Adaptive Attacks. Tab. 4 shows results of adaptive attacks with perturbation budget $\epsilon = 4/255$. Our defense successfully reduces the average data size by 67.75% and 67.62% for low frequency attack and regional attack respectively while the accuracy only decreases 1.21% and 1.41% respectively. **Thus, the proposed approach is resilient to adaptive attacks.**

6 Related Work

Distributed Deep Neural Network. Distributed DNN is proposed to meet the challenge of deploying artificial intelligence to resource-constrained platforms such as mobile devices and Internet of Things (IoT) devices [41]. To minimize the end-to-end latency across devices, dimension reduction designs (*i.e.*, *bottlenecks*) are introduced to compress the data size of intermediate representations that need to be sent [18,54].

However, unlike autoencoders [32,36,49] that have symmetric designs on both encoder and decoder sides, distributed DNNs usually have asymmetric designs due to the limit of computation resources, and thus inject the *bottlneck* in early layers [18,38–40,54]. As a result, naively end-to-end trained distributed DNNs have noticeable performance loss [18,54]. To preserve accuracy, [38] proposed to use knowledge distillation to train the distributed DNN separately while [39] proposed a multi-stage training approach for each part of the DNN.

Along with *bottlneck*-based distributed DNNs, different coding-based approaches are also applied to reduce the data size of the latent representations. [2] proposed to apply codec-based compression such as JPEG to latent representations and [15] adopted spatio-temporal coding such as HEVC to compress the streaming latent features in video tasks. [42,55,59] use entropy coding to achieve a higher compression ratio of latent representations.

Entropy Coding in Deep Data Compression. [5] first introduced a fully factorized prior to integrate entropy coding with variational autoencoder, showcasing superior quality enhancement over conventional codec-based methods like JPEG for image compression. Building upon this, [6] extended the conventional factorized prior model to a learnable hierarchical hyper prior, resulting in improved performance. [43] proposed a joint auto-regressive and hyper prior design while [44] proposed a channel-aware entropy coding scheme. In addition, [14] introduced the attention mechanism to prior models and [1] extended the technique to video compression.

To minimize the communication overhead in distributed DNN scenarios, [55] for the first time adopted the entropy coding to further compress the latent representations after the *bottleneck* while [59] extended the approach to object detection. However, these approaches overlooked resource constraints, resulting in over-complex network designs for mobile devices. Conversely, [42] integrated entropy coding with a multi-stage training strategy [39], aiming to optimize a tripartite trade-off encompassing computation resources in mobile devices, communication overhead between mobile and edge devices, and end-to-end performance of neural networks.

Resilience of DNN efficiency. While the reliability of DNNs have been extensively investigated, limited attention has been devoted to interference that undermine their efficiency. [22] explored this direction by demonstrating that imperceptible perturbations to input, leveraging intermediate representations of DNNs, could nullify the computation savings achieved by dynamic depth neural networks, which adapt their depth based on input complexity. [46] extended these attacks to target both dynamic depth and dynamic width neural networks. Moreover, in [45], the authors proposed simultaneous adjustments to the direction and magnitude of attacks, enhancing their effectiveness. [27] revealed that early exit dynamic DNNs could be tricked into late-stage inferences, significantly slowing down inference speed.

The study of efficiency vulnerabilities in DNNs has also ventured into real-world deployment scenarios. [33] focused on attacking LiDAR-based detection systems, introducing latency in detections and exposing vulnerabilities in critical contexts such as self-driving cars. [13] demonstrated that neural image caption generation models could be manipulated to incur increased computation costs by inducing unnecessary decoder calls for token generation.

In star contrast to existing research, our work investigates a novel threat emerging in distributed DNNs. Here, adversaries not only compromise the communication efficiency of entropy-coded distributed DNNs but also pose a threat to other users by saturating the transmission bandwidth.

7 Conclusion

This paper has investigated the resilience of entropy models in distributed DNNs against both intentional and unintentional interference. We conducted thorough evaluations using 3 different DNN architectures, 2 entropy model designs, and 4 rate-distortion trade-off factors with common corruption datasets and adversarial attacks. Our analysis disentangled compression features in both spatial and frequency domains, revealing vulnerabilities of the entropy model to specific types of perturbations. Building on these findings, we proposed a standalone defense strategy aimed at reducing data size with minimal task-oriented performance loss. Our future work will focus on designing more advanced defense approaches for distributed DNNs that are resilient in both compression and classification tasks.

Acknowledgements

This work is funded in part by the Air Force Office of Scientific Research under contract number FA9550-23-1-0261, by the Office of Naval Research under contract number N00014-23-1-2221, as well as by National Science Foundation grants CNS-2134973 and CNS-2312875. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

References

1. Agustsson, E., Minnen, D., Johnston, N., Balle, J., Hwang, S.J., Toderici, G.: Scale-space flow for end-to-end optimized video compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8503–8512 (2020)
2. Alvar, S.R., Bajić, I.V.: Pareto-optimal bit allocation for collaborative intelligence. IEEE Transactions on Image Processing **30**, 3348–3361 (2021)
3. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European conference on computer vision. pp. 484–501. Springer (2020)
4. Arvinte, M., Cornelius, C., Martin, J., Himayat, N.: Investigating the adversarial robustness of density estimation using the probability flow ode. arXiv preprint arXiv:2310.07084 (2023)
5. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: 5th International Conference on Learning Representations, ICLR 2017 (2017)
6. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. In: International Conference on Learning Representations (2018)
7. Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. Advances in neural information processing systems **32** (2019)
8. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6541–6549 (2017)
9. Boigné, E., Parkinson, D.Y., Ihme, M.: Towards data-informed motion artifact reduction in quantitative ct using piecewise linear interpolation. IEEE Transactions on Computational Imaging **8**, 917–932 (2022)
10. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM workshop on artificial intelligence and security. pp. 3–14 (2017)
11. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. Ieee (2017)
12. Chen, J., Jordan, M.I., Wainwright, M.J.: Hopskipjumpattack: A query-efficient decision-based attack. In: 2020 ieee symposium on security and privacy (sp). pp. 1277–1294. IEEE (2020)
13. Chen, S., Song, Z., Haque, M., Liu, C., Yang, W.: Nicgslowdown: Evaluating the efficiency robustness of neural image caption generation models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15365–15374 (2022)

14. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7939–7948 (2020)
15. Choi, H., Bajić, I.V.: Deep feature compression for collaborative object detection. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 3743–3747. IEEE (2018)
16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
17. Duda, J.: Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding. arXiv preprint arXiv:1311.2540 (2013)
18. Eshratifar, A.E., Esmaili, A., Pedram, M.: Bottlenet: A deep learning architecture for intelligent mobile cloud computing services. In: 2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). pp. 1–6. IEEE (2019)
19. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2018)
20. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
21. Guo, C., Frank, J.S., Weinberger, K.Q.: Low frequency adversarial perturbation. arXiv preprint arXiv:1809.08758 (2018)
22. Haque, M., Chauhan, A., Liu, C., Yang, W.: Ilfo: Adversarial attack on adaptive neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14264–14273 (2020)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
24. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8349 (2021)
25. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2018)
26. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
27. Hong, S., Kaya, Y., Modoranu, I.V., Dumitraş, T.: A panda? no, it's a sloth: Slowdown attacks on adaptive multi-exit neural network inference. arXiv preprint arXiv:2010.02432 (2020)
28. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1314–1324 (2019)
29. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
30. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: International conference on machine learning. pp. 2137–2146. PMLR (2018)

31. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2704–2713 (2018)
32. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
33. Liu, H., Wu, Y., Yu, Z., Vorobeychik, Y., Zhang, N.: Slowlidar: Increasing the latency of lidar-based detection using adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5146–5155 (2023)
34. Liu, P., Qi, B., Banerjee, S.: Edgeeye: An edge service framework for real-time intelligent video analytics. In: Proceedings of the 1st international workshop on edge systems, analytics and networking. pp. 1–6 (2018)
35. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
36. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21. pp. 52–59. Springer (2011)
37. Matsubara, Y., Yang, R., Levorato, M., Mandt, S.: Sc2 benchmark: Supervised compression for split computing. Transactions on machine learning research (2023)
38. Matsubara, Y., Callegaro, D., Baidya, S., Levorato, M., Singh, S.: Head network distillation: Splitting distilled deep neural networks for resource-constrained edge computing systems. IEEE Access **8**, 212177–212193 (2020)
39. Matsubara, Y., Callegaro, D., Singh, S., Levorato, M., Restuccia, F.: Bottlefit: Learning compressed representations in deep neural networks for effective and efficient split computing. In: 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). pp. 337–346. IEEE (2022)
40. Matsubara, Y., Levorato, M.: Neural compression and filtering for edge-assisted real-time object detection in challenged networks. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 2272–2279. IEEE (2021)
41. Matsubara, Y., Levorato, M., Restuccia, F.: Split computing and early exiting for deep learning applications: Survey and research challenges. ACM Computing Surveys **55**(5), 1–30 (2022)
42. Matsubara, Y., Yang, R., Levorato, M., Mandt, S.: Supervised compression for resource-constrained edge computing systems. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2685–2695 (2022)
43. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. Advances in neural information processing systems **31** (2018)
44. Minnen, D., Singh, S.: Channel-wise autoregressive entropy models for learned image compression. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 3339–3343. IEEE (2020)
45. Pan, J., Foo, L.G., Zheng, Q., Fan, Z., Rahmani, H., Ke, Q., Liu, J.: Gradmdm: Adversarial attack on dynamic networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

46. Pan, J., Zheng, Q., Fan, Z., Rahmani, H., Ke, Q., Liu, J.: Gradauto: Energy-oriented attack on dynamic neural networks. In: European Conference on Computer Vision. pp. 637–653. Springer (2022)
47. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing network design spaces. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10428–10436 (2020)
48. Ran, X., Chen, H., Zhu, X., Liu, Z., Chen, J.: Deepdecision: A mobile deep learning framework for edge video analytics. In: IEEE INFOCOM 2018-IEEE conference on computer communications. pp. 1421–1429. IEEE (2018)
49. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. In: Proceedings of the 28th international conference on international conference on machine learning. pp. 833–840 (2011)
50. Rissanen, J., Langdon, G.: Universal modeling and coding. IEEE Transactions on Information Theory **27**(1), 12–23 (1981)
51. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena **60**(1-4), 259–268 (1992)
52. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
53. Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal **27**(3), 379–423 (1948)
54. Shao, J., Zhang, J.: Bottlenet++: An end-to-end approach for feature compression in device-edge co-inference systems. In: 2020 IEEE International Conference on Communications Workshops (ICC Workshops). pp. 1–6. IEEE (2020)
55. Singh, S., Abu-El-Haija, S., Johnston, N., Ballé, J., Shrivastava, A., Toderici, G.: End-to-end learning of compressible features. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 3349–3353. IEEE (2020)
56. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: 2nd International Conference on Learning Representations, ICLR 2014 (2014)
57. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2820–2828 (2019)
58. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. Advances in Neural Information Processing Systems **32** (2019)
59. Yuan, Z., Rawlekar, S., Garg, S., Erkip, E., Wang, Y.: Feature compression for rate constrained object detection on the edge. In: 2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR). pp. 1–6. IEEE (2022)
60. Zhang, W., Zhang, Z., Zeadally, S., Chao, H.C., Leung, V.C.: Masm: A multiple-algorithm service model for energy-delay optimization in edge artificial intelligence. IEEE Transactions on Industrial Informatics **15**(7), 4216–4224 (2019)