

# End-to-End Image Classification and Compression With Variational Autoencoders

Lahiru D. Chamain<sup>1b</sup>, *Student Member, IEEE*, Siyu Qi<sup>1b</sup>, and Zhi Ding<sup>1b</sup>, *Fellow, IEEE*

**Abstract**—The past decade has witnessed the rising dominance of deep learning and artificial intelligence in a wide range of applications. In particular, the ocean of wireless smartphones and IoT devices continue to fuel the tremendous growth of edge/cloud-based machine learning (ML) systems, including image/speech recognition and classification. To overcome the infrastructural barrier of limited network bandwidth in cloud ML, existing solutions have mainly relied on traditional compression codecs such as JPEG that were historically engineered for human-end users instead of ML algorithms. Traditional codecs do not necessarily preserve features important to ML algorithms under limited bandwidth, leading to potentially inferior performance. This work investigates application-driven optimization of programmable commercial codec settings for networked learning tasks such as image classification. Based on the foundation of variational autoencoders (VAEs), we develop an end-to-end networked learning framework by jointly optimizing the codec and classifier without reconstructing images for a given data rate (bandwidth). Compared with the standard JPEG codec, the proposed VAE joint compression and classification framework achieves classification accuracy improvement by over 10% and 4%, respectively, for CIFAR-10 and ImageNet-1k data sets at data rate of 0.8 bpp. Our proposed VAE-based models show 65%–99% reductions in encoder size,  $\times 1.5$ – $\times 13.1$  improvements in inference speed, and 25%–99% savings in power compared to baseline models. We further show that a simple decoder can reconstruct images with sufficient quality without compromising classification accuracy.

**Index Terms**—Classification, compression, end-to-end, reconstruction, variational autoencoders (VAEs).

## I. INTRODUCTION

THE CONCEPT “Internet of Everything (IoE)” generalizes the idea of Internet of Things (IoT) to a broader paradigm. IoE nodes are connected by networks capable of communicating data generated from sensing and processing [1]. The power of IoE has already bolstered learning-intensive technologies, such as self-driving cars, surveillance cameras, smart cities, and smart transportation where deep learning (DL) applications are increasingly integrated with network services. As part of the networked learning paradigm, these DL applications together with network

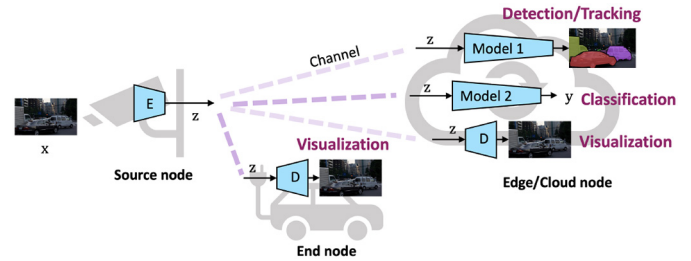


Fig. 1. Networked ML instance: video/image frames from a camera are sent to a server for vehicle detection, traffic monitoring, and used by vehicles with an obstructed view for assisted/self-driving-related visualizations/decision-making. E: Encoder and D: Decoder.

services facilitate machine-to-machine and machine-to-cloud server communications [2]–[6].

One such example of networked learning is illustrated in Fig. 1. In automated inference systems, such as pedestrian detection or object tracking for applications like self-driving, media data (image or video) collected by a source node is transported to remote network cloud nodes to carry out more complex inference processes. The data transmission can be for multiple inference tasks (such as vehicle detection, traffic monitoring, or risk warning) at the cloud servers or visualization purposes at cloud or end nodes. For instance, video/image frames captured by a street camera are sent to a server for vehicular detection, crowd activity monitoring, etc., and used by smart vehicles with an obstructed view for assisted/self-driving-related visualizations/decision making.

Given the explosive growth and deployment of IoT devices [7] and massive data collected by such devices [8], traditional cloud computing struggles to keep up with the demands of large IoT networks. Typical IoT configuration features source devices only as data collectors while cloud nodes are responsible for processing and analysis, hence, is limited by network link capacity, delay and losses, leading to long latency, unreliable inference, and scalability issues [9]–[11]. Edge computing addresses these issues by deploying computing services in proximity to IoT devices [12]. Many image and video sensing devices, such as smart phones, vehicular sensors, and home/street cameras have sufficient resources to collect data and perform preliminary preprocessing and feature extraction, but not enough to implement full machine learning (ML) algorithms on-device [10]. With this new configuration of networked AI coupled with edge computing, such source devices can share a part of the computational burden by performing

Manuscript received 7 October 2021; revised 31 January 2022 and 16 May 2022; accepted 6 June 2022. Date of publication 13 June 2022; date of current version 24 October 2022. This work was supported by the National Science Foundation under Grant 2002927 and Grant 2002937. (Corresponding author: Lahiru D. Chamain.)

The authors are with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA 95616 USA (e-mail: hchamain@ucdavis.edu; syqi@ucdavis.edu; zding@ucdavis.edu).

Digital Object Identifier 10.1109/JIOT.2022.3182313

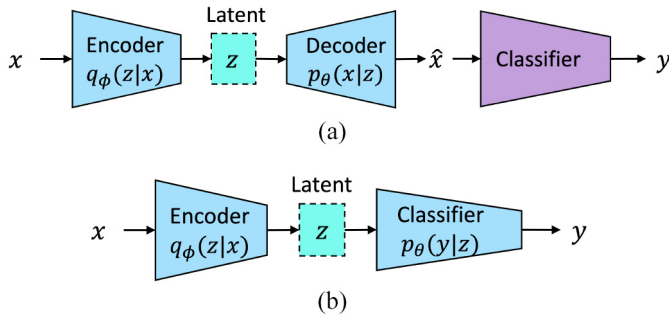


Fig. 2. (a) VAE-based reconstruction before classification: Encoder transforms source image  $x$  to low dimensional  $z$ . Decoder reconstructs image  $\hat{x}$  based on  $z$ . (b) Proposed joint compression and classification: Without image reconstruction, classifier directly generates class label  $y$  based on  $z$ .

preprocessing and feature extraction, and deliver the extracted features to more powerful and resource-rich edge/fog nodes for complex ML tasks.

When designing communication networks specifically targeting networked AI over edge/cloud for above such time-sensitive tasks, source data transmitted to the edge/cloud may fulfill two main requirements. On the one hand, the data transport must achieve high coding efficiency ensuring low power usage, low latency, and high bandwidth efficiency. On the other hand, source data compression for transport must maintain high learning (task) accuracy. Therefore, there is a clear design tradeoff between source coding rate and learning accuracy. This work designs a variational autoencoder (VAE)-based compression and learning framework to achieve a better coding rate and learning accuracy tradeoff ensuring low power usage, low latency, and low memory usage.

Our focused investigation in this work is on an efficient image codec for classification applications. Conventional classification applications begin by reconstructing images of interest at the decoder before inference [13]–[15]. See Fig. 2(a). We note the following two observations where the reconstruction can introduce inefficiency in cloud-based AI applications. First, when performing classification inference on the images compressed with image codecs that are originally optimized for rate-distortion performance targeting visualization applications, the classification accuracy suffers at high compression ratios under limited data rate [16]–[20]. Second, inference on images compressed with standard compression codecs, such as JPEG2000 demonstrates non-negligible inference speed and accuracy degradation compared to with end-to-end-optimized joint classification and compression frameworks [17], [18] that bypass image reconstruction at decoders. Motivated by these observations, the proposed VAE-based classifier bypasses the reconstruction and is designed to naturally support end-to-end optimization for rate-accuracy performance. See Fig. 2(b).

In terms of device complexity, conventional video/image compression codecs, such as VVC [21], HEVC [22], and JPEG [23] featuring a more complex encoder and a simpler decoder could be less desirable on low cost source devices. Instead, a more desirable data compression and learning model for massive deployment in a network-AI edge/cloud

should feature a simple encoder with low power and memory usage and a relatively more complex decoding and learning edge/cloud node. The trained encoder can reside in source devices with low computational power and memory to encode and compress data collections. Further, they can selectively compress and transmit task-important underlying features (e.g., latent) of the raw source data to other edges/cloud servers dedicated for decoding and learning ensuring rate-accuracy performance.

The approaches in [16]–[18] are likely suboptimal in terms of coding efficiency by optimizing only a part of pre-engineered JPEG2000 encoding-decoding pipeline. Addressing this, as part of end-to-end optimization of the codec, some recent DL-based image/video compression codecs showcase a learning-based encoder (autoencoder) that can be optimized to minimize a given loss function [20], [24]–[28]. They demonstrated the importance of learning code word distribution for efficient compression instead of using fixed code word tables as in JPEG. However, the image encoders used in the above works are still high in complexity, demanding high memory, and power and resulting in lower inference speeds. Hence, they are not suitable for broad IoT deployment. Thus, we propose to use light-weight encoders and to model code word distributions for VAE-based classifier achieving high coding efficiency.

To achieve the aforementioned objectives in a networked AI environment, we propose a VAE-based joint compression and classification model [shown in Fig. 2(b)] that enables learning on the latent feature space to efficiently encode/compress and effectively classify images through end-to-end training. We aim to achieve high accuracy, fast inference, low bandwidth usage, and low latency over network links. Furthermore, our proposed framework features simple encoders while ensuring reusability of the transmitted features.

Our two main contributions of this work are listed as follows.

- 1) A new information-theoretical formulation of a VAE-based end-to-end, joint compression, and classification framework.
- 2) Development of power saving, low-complexity-encoder, and faster-inference-classifier for edge/cloud-based networked image classification applications.

Furthermore, because encoded images may serve the dual-use of human visual perception and ML in certain practical applications, we investigate the feasibility of adapting the VAE-based classifier as a dual-use codec for efficiently encoding image features delivered to the cloud for both accurate image classification and image reconstruction of sufficiently high quality (PSNR).

We organize this manuscript as follows. Section II presents the basic problem formulation and review the concept of VAEs. Section III introduces our new proposal of a VAE-based end-to-end solution for classification by describing the basic learning model and the rate-accuracy function for joint classification compression. In Section IV we describe our experiment setup and present our test results. Additionally, Section V presents benchmark comparison of model complexity, inference speed, and power consumption for our proposed

models. In Section VI, we discuss how to adapt the proposed models for joint classification and reconstruction. Section VII further provides theoretical basis that connects the proposed VAE-based approaches to frameworks such as information bottleneck (IB) [29], [30].

## II. RELATED WORKS

### A. End-to-End Learning Framework

Originally designed to target remote human vision applications over resource constrained channels, popular image compression codecs, such as JPEG [31] and JPEG2000 [32] feature an engineered data processing workflow. This workflow mainly consists of a level offset, a color transformation (e.g., RGB to YCbCr), a domain transformation, such as discrete cosine transformation (DCT) or discrete wavelet transformation (DWT), a quantizer and a block encoder (e.g., Huffman or Arithmetic) to convert the quantized transform domain coefficients to a bit-stream for transmission.

These engineered blocks of conventional image codecs can often achieve good reconstruction results for a given data rate. Similarly, for a given image quality/distortion, such codecs can generate low-rate bit-streams for transmission over low bandwidth network channels. However, these popular image/video codecs, such as JPEG and JPEG2000 are not optimized for remote learning tasks. Thus, a naïve and direct adoption of these codecs for cloud-based learning applications do not guarantee good performance or high spectrum efficiency. Conceptually, source codecs should be customized and optimized for specific cloud-based learning tasks [18], [20]. Therefore, our objective in this work aims to optimize rate-accuracy performance in terms of encoding/compression rate and classification accuracy.

### B. Source Data Reconstruction With VAEs

When designing codecs for source compression to achieve optimized rate-distortion performance, accurate modeling of the data distribution is of vital importance. Depending on the underlying input distribution, the process of “generative modeling” can be challenging. Often, the input data ( $\mathcal{X}$ ) manifold is high dimensional and is complex to characterize.

Several likelihood-based methods have been proposed in the literature for generative modeling, such as auto-regressive models [33], flow-based methods [34], [35], and VAEs [36]. In this work, we leverage the concept of VAE for joint classification and compression. Without relying on strong assumptions, VAE exhibits fast training with back-propagation [37]. Such property is advantageous in comparison with model-based approaches relying on strong assumptions or requiring high computation complexity such as the Markov chain Monte Carlo (MCMC) [36]. Further, VAE-based models are amenable to naturally interpretable loss terms that are directly related to rate-distortion tradeoff in lossy compression [38], as shown later.

We can capture the general concept of VAE via Fig. 2(a). A VAE consists of an encoder for mapping the high-dimensional input  $\mathbf{x}$  into a latent representation  $\mathbf{z}$ , followed by a decoder in charge of reconstructing the input that is

TABLE I  
SUMMARY OF NOTATIONS

Notation	Meaning
$\mathcal{X}$	set of input images
$\mathbf{x}$	input image
$\hat{\mathbf{x}}$	reconstructed image
$y_{\text{gt}}$	ground truth label of image $\mathbf{x}$
$y$	estimated label of image $\mathbf{x}$
$\mathbf{z}$	latent vector representation of $\mathbf{x}$
$\hat{\mathbf{z}}$	quantized $\mathbf{z}$
$d_\ell$	quantization level
$\mathcal{D}$	set of quantization levels $d_\ell$
$p_{\mathbf{x},\theta}$	probability distribution of $\mathbf{x}$ parameterized by $\theta$
$p_{\mathbf{z} \mathbf{x},\theta}$	conditional distribution of $\mathbf{z}$ parameterized by $\theta$
$q_{\mathbf{z} \mathbf{x},\phi}$	approximated distribution of $p_{\mathbf{z} \mathbf{x},\theta}$ parameterized by $\phi$
$\hat{p}_\theta(y \mathbf{x})$	estimated class label distribution with a model parameterized by $\theta$ given the latent $\mathbf{x}$
$\eta_\theta(y \mathbf{z})$	estimated class label distribution with a model parameterized by $\theta$ given the latent $\mathbf{z}$
$P_{i,\ell_i}(\hat{\mathbf{z}} \mathbf{x})$	conditional probability of the latent value $\hat{z}_i$ that takes the value of the quantization level $d_{\ell_i}$
$Q_{i,\ell_i}(y_{\text{gt}} \hat{\mathbf{z}})$	class label probability for each $\hat{z}_i$ that takes the value of $d_{\ell_i}$
$\text{KL}(p, q)$	KL divergence between distributions $p$ and $q$
$\text{CE}(y_{\text{gt}}, y)$	cross entropy between distribution of $y_{\text{gt}}$ and $y$
$H(p_{\mathbf{x},\theta})$	entropy of variable $\mathbf{x}$ with distribution $p_{\mathbf{x},\theta}$
$\beta, \gamma$	trade-off coefficients for rate, distortion.

denoted as input estimate  $\hat{\mathbf{x}}$ . The encoder’s output  $\mathbf{z}$  is a low-dimensional latent vector representing distinct features from the input data  $\mathbf{x}$ . The encoder functionality is to compress and to extract critical features by mapping input  $\mathbf{x}$  into  $\mathbf{z}$ . The decoder can rely on  $\mathbf{z}$  to reconstruct  $\mathbf{x}$  for various remote applications. Thus, the VAE encoder makes it possible to store and transmit  $\mathbf{z}$  instead of  $\mathbf{x}$  to preserve bandwidth and storage.

In the VAE formulation, the observable input data set  $\mathcal{X}$  is assumed to consist of i.i.d. samples of  $\mathbf{x}$  which are generated by some random processes that involves an unobserved random variable  $\mathbf{z}$  and generative model parameters  $\theta$  via

$$p_{\mathbf{x},\theta}(\mathbf{x}) = \int p_{\mathbf{z},\theta}(\mathbf{z})p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})d\mathbf{z}. \quad (1)$$

Neither the true parameters  $\bar{\theta}$  nor  $\mathbf{z}$  are known.

Within this framework [36], the encoder can be viewed as to provide an efficient posterior estimate of the latent vector  $\mathbf{z}$  from an observed input  $\mathbf{x}$  for a given parameter setting  $\theta$ . The decoder provides an approximate marginal inference of  $\mathbf{x}$  upon reception of the latent vector  $\mathbf{z}$  from the encoding transmitter.

### C. Variational Bound for Reconstruction

VAE considers the general case when the posterior  $p_{\mathbf{z}|\mathbf{x},\theta}(\mathbf{z}|\mathbf{x})$  is intractable for which an approximation  $q_{\mathbf{z}|\mathbf{x},\phi}(\mathbf{z}|\mathbf{x})$  parameterized by  $\phi$  is introduced to act as the encoder. For an arbitrary distribution  $q_{\mathbf{z}|\mathbf{x},\phi}(\mathbf{z}|\mathbf{x})$ , we can derive the “variational bound” that follows the outlines of [36] and [37] with detailed proof below for both reconstruction and classification. Throughout the presentation, we denote an instance of the random variables  $\mathbf{x}$  as  $\mathbf{x}$ . Further in Table I, we summarize

the notations used in the current work. Assume  $\mathbf{x} \in \mathcal{X}$  is a random sample of the random variable  $\mathbf{x}$  which follows a generative model parameterized by  $\theta$  from an unobserved random variable  $\mathbf{z}$ . We can write

$$p_{\mathbf{x},\theta}(\mathbf{x}) = \int p_{\mathbf{z},\theta}(\mathbf{z})p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})d\mathbf{z}. \quad (2)$$

For an arbitrary distribution  $q_{\mathbf{z},\phi}(\mathbf{z})$ , we can write

$$\begin{aligned} p_{\mathbf{x},\theta}(\mathbf{x}) &= \int p_{\mathbf{z},\theta}(\mathbf{z})p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})d\mathbf{z} \\ &= \int p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})p_{\mathbf{z},\theta}(\mathbf{z})\frac{q_{\mathbf{z},\phi}(\mathbf{z})}{q_{\mathbf{z},\phi}(\mathbf{z})}d\mathbf{z} \\ &= E_{q_{\mathbf{z},\phi}}\left[p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})\frac{p_{\mathbf{z},\theta}(\mathbf{z})}{q_{\mathbf{z},\phi}(\mathbf{z})}\right]. \end{aligned} \quad (3)$$

Taking  $-\log$  of both sides leads to

$$-\log p_{\mathbf{x},\theta}(\mathbf{x}) = -\log E_{q_{\mathbf{z},\phi}}\left[p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})\frac{p_{\mathbf{z},\theta}(\mathbf{z})}{q_{\mathbf{z},\phi}(\mathbf{z})}\right]. \quad (4)$$

Applying Jensen's inequality to (4), we write

$$\begin{aligned} -\log p_{\mathbf{x},\theta}(\mathbf{x}) &\leq E_{q_{\mathbf{z},\phi}} - \log\left[p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})\frac{p_{\mathbf{z},\theta}(\mathbf{z})}{q_{\mathbf{z},\phi}(\mathbf{z})}\right] \\ &\leq -E_{q_{\mathbf{z},\phi}} \log p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z}) \\ &\quad - E_{q_{\mathbf{z},\phi}} \log\left[\frac{p_{\mathbf{z},\theta}(\mathbf{z})}{q_{\mathbf{z},\phi}(\mathbf{z})}\right]. \end{aligned} \quad (5)$$

Following the definition of KL divergence, we write:

$$-\log p_{\mathbf{x},\theta}(\mathbf{x}) \leq -E_{q_{\mathbf{z},\phi}} \log p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z}) + \text{KL}(q_{\mathbf{z},\phi}||p_{\mathbf{z},\theta}). \quad (6)$$

Since  $q_{\mathbf{z},\phi}(\mathbf{z})$  is arbitrary, we can replace  $q_{\mathbf{z},\phi}(\mathbf{z})$  with conditional density  $q_{\mathbf{z}|\mathbf{x},\phi}(\mathbf{z}|\mathbf{x})$  and write [36], [37]

$$\begin{aligned} -\log p_{\mathbf{x},\theta}(\mathbf{x}) &\leq -E_{q_{\mathbf{z}|\mathbf{x},\phi}}[\log p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})] \\ &\quad + \text{KL}(q_{\mathbf{z}|\mathbf{x},\phi}||p_{\mathbf{z},\theta}) \end{aligned} \quad (7)$$

where KL denotes the Kullback–Leibler divergence.

This variational bound can serve as an optimization surrogate when direct minimization of  $-\log p_{\mathbf{x},\theta}(\mathbf{x})$  is intractable. Functionally,  $q_{\mathbf{z}|\mathbf{x},\phi}$  models the probabilistic encoder and  $p_{\mathbf{x}|\mathbf{z},\theta}$  models the reconstruction decoder as in Fig. 2(a). The first RHS term in (7) is the conditional entropy of  $\mathbf{x}$  given  $\mathbf{z}$  which quantifies the reconstruction loss. The second RHS term  $\text{KL}(q_{\mathbf{z}|\mathbf{x},\phi}||p_{\mathbf{z},\theta})$  is related to the coding cost of the latents as shown later. Hence, the variational bound can be utilized to form the classical variational loss for reconstruction [36]

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = E_{q_{\mathbf{z}|\mathbf{x},\phi}}[-\log p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})] + \text{KL}(q_{\mathbf{z}|\mathbf{x},\phi}||p_{\mathbf{z},\theta}). \quad (8)$$

When minimizing the variational bound as the loss function given in (8) that consists of two parts, a hard constraint can be imposed on the coding cost  $\text{KL}(q_{\mathbf{z}|\mathbf{x},\phi}||p_{\mathbf{z},\theta})$  while minimizing the reconstruction loss. To this extent, Higgins *et al.* [39] further suggested adding a tradeoff parameter  $\beta(\geq 1)$  to (8) to reformulate a  $\beta$ -VAE loss function also adopted in [14], [27], and [38] as

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = E_{q_{\mathbf{z}|\mathbf{x},\phi}}[-\log p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z})] + \beta \text{KL}(q_{\mathbf{z}|\mathbf{x},\phi}||p_{\mathbf{z},\theta}). \quad (9)$$

Note that we can leverage the definition of cross entropy (CE) to rewrite

$$\text{KL}(q_{\mathbf{z}|\mathbf{x},\phi}||p_{\mathbf{z},\theta}) = \text{CE}(q_{\mathbf{z}|\mathbf{x},\phi}, p_{\mathbf{z},\theta}) - H(q_{\mathbf{z}|\mathbf{x},\phi}).$$

The term  $\text{CE}(q_{\mathbf{z}|\mathbf{x},\phi}, p_{\mathbf{z},\theta})$  averages the entropy  $-\log(p_{\mathbf{z},\theta})$  over the encoder distribution  $q_{\mathbf{z}|\mathbf{x},\phi}$  and captures the average encoding “cost” of latent representation  $\mathbf{z}$ . Following the approach of [38], if we consider only the deterministic encoder  $\mathbf{z} = \phi(\mathbf{x})$  for which  $q_{\mathbf{z}|\mathbf{x},\phi}(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - \phi(\mathbf{x}))$  and  $H(q_{\mathbf{z}|\mathbf{x},\phi}) = 0$ , we have

$$\begin{aligned} \text{KL}(q_{\mathbf{z}|\mathbf{x},\phi}||p_{\mathbf{z},\theta}) &= \text{CE}(q_{\mathbf{z}|\mathbf{x},\phi}, p_{\mathbf{z},\theta}) \\ &= \text{CE}(\delta(\mathbf{z} - \phi(\mathbf{x})), p_{\mathbf{z},\theta}(\mathbf{z})) \\ &= -\log p_{\mathbf{z},\theta}(\phi(\mathbf{x})). \end{aligned} \quad (10)$$

From (10), the  $\beta$ -VAE loss function for a deterministic encoder can be expressed as [38]

$$\begin{aligned} \mathcal{L}_{\theta,\phi}(\mathbf{x}) &= E_{q_{\mathbf{z}|\mathbf{x},\phi}}[-\log p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\mathbf{z}) - \beta \log p_{\mathbf{z},\theta}(\mathbf{z})] \\ &= -\log p_{\mathbf{x}|\mathbf{z},\theta}(\mathbf{x}|\phi(\mathbf{x})) - \beta \log p_{\mathbf{z},\theta}(\phi(\mathbf{x})). \end{aligned} \quad (11)$$

When  $\beta = 1$ , (11) corresponds to the VAE loss function in (9). Setting  $\beta \geq 1$  can impose a hard constraint on the latent representation [39] to limit the coding cost. In short, the approximated  $\beta$ -VAE loss is the sum of the reconstruction loss and the cost of encoding  $\mathbf{z}$  weighted by  $\beta$ . The weight  $\beta$  facilitates a rate-distortion tradeoff parameter. One extreme case would be to encode nothing, i.e.,  $\mathbf{z} = 0$ , in which case the encoding cost is zero whereas the reconstruction loss from  $\mathbf{z} = 0$  would be gigantic. The other extreme case would be to encode  $\mathbf{x}$  directly, i.e.,  $\mathbf{z} = \mathbf{x}$ , in which case the reconstruction loss is 0 whereas coding cost would be high.

#### D. VAE-Based Classifier

When reviewing the VAE framework, data reconstruction from  $\mathbf{z}$  is typically the learning objective. We note, however, that for automated learning tasks such as image classification, reconstruction of  $\mathbf{x}$  may not always be the end goal [20], [40] and may be unnecessary in many cases. For example, a practical security-related automated object detection system would rely on AI or DL algorithm trained on the latent vector  $\mathbf{z}$  for object detection and/or recognition. Hence, the reconstruction of image frame  $\hat{\mathbf{x}}$  is unnecessary for inference. Only at rare occasions, such as evidence collection or inspection, the system may have to recover RGB frames for human visualization. Thus, the image reconstruction step can be performed only when required from the stored latent vectors.

Considering these practical AI/learning applications, we propose a VAE-based classifier by removing the VAE decoder in Fig. 2(a) to directly connect the latent code  $\mathbf{z}$  with a data classifier to perform the learning task, as shown in Fig. 2(b). In this framework, variable  $y$  denotes the label of the output class based on  $\mathbf{z}$ . Our framework aims to jointly optimize the encoder and the classifier with end-to-end training.

Before presenting details of our proposed VAE framework, we first examine some related recent works on the joint learning of compression and classification based on autoencoders. Among others, one approach constructs a loss function by



combining the latent entropy  $H(\mathbf{z})$  with the classification CE loss  $\mathcal{L}_{\text{CE}}$  resulting from classification  $v_0(\hat{\mathbf{x}})$  according to the reconstructed  $\hat{\mathbf{x}}(\mathbf{z})$  from  $\mathbf{z}$  [as shown in Fig. 2(a)]

$$\mathcal{L}(\mathbf{x}, y_{\text{gt}}) = \mathcal{L}_{\text{CE}}(y_{\text{gt}}, v_0(\hat{\mathbf{x}}(\mathbf{z})) + \beta H(\mathbf{z}|\mathbf{x}). \quad (12)$$

Here,  $y_{\text{gt}}$  denoted the ground truth class of image sample  $\mathbf{x}$ . Clearly, this approach requires a reconstruction step prior to classification [41], [42].

Another approach is to construct the loss function by combining the latent entropy  $H(\mathbf{z})$  with the classification CE loss-based directly on the latent representation  $\mathbf{z}$ . Hence, the loss considering the classification  $v_1(\mathbf{z})$  directly based on encoded  $\mathbf{z}(\mathbf{x})$  is

$$\mathcal{L}(\mathbf{x}, y_{\text{gt}}) = \mathcal{L}_{\text{CE}}(y_{\text{gt}}, v_1(\mathbf{z}(\mathbf{x}))) + \beta H(\mathbf{z}|\mathbf{x}). \quad (13)$$

Compared with the first approach, this formulation can potentially generate faster and more efficient training/inference models by skipping the reconstruction step [43]. For example, Chamain *et al.* [16], [18] introduced an image classification model by jointly training to optimize both the JPEG2000 encoder and the CNN classifier without image reconstruction. While achieving faster inference and higher accuracy, there is limited improvement since the proposed end-to-end framework only optimizes the quantization block of the encoder. Further, the majority of existing VAE-based image classification frameworks either still perform this unnecessary reconstruction step [13]–[15] or neglect the bandwidth aspect of the latent representation [44] making them impractical for wide IoT deployment in cloud-based image classification.

### III. NEW VAE CLASSIFICATION FRAMEWORK

In this work, we focus on the second approach described in (13) and develop a novel VAE model for joint compression and classification encoding. Specifically, we design a VAE-based classifier for direct image classification based on the latent vectors without image reconstruction, as seen from Fig. 2. Targeting cloud-based classification over bandwidth-limited network data links, we implement adaptive entropy coders with context modeling aimed at improving the overall rate-accuracy performance tradeoff. Further, we demonstrate that spatial domain RGB images can also be reconstructed for practical applications by fine-tuning a separate decoder under the proposed classification setting.

#### A. Variational Bound and Loss for Classification

The performance of a classifier can be measured by CE loss from labeling. Consider an i.i.d. data sample  $\mathbf{x}$  and corresponding label  $y$  that belongs to a set  $\mathcal{Y}$  of unique labels, with  $\{(\mathbf{x}, y) | \mathbf{x} \in \mathcal{D}, y \in \mathcal{Y}\}$ . We can denote  $\eta_{\theta}(y|\mathbf{z})$  as the estimated label distribution with a model parameterized by  $\theta$  given the latent  $\mathbf{z}$ . For a latent  $\mathbf{z}$  mapped from given  $\mathbf{x}$ , the estimated label distribution is simply

$$\hat{\rho}_{\theta}(y|\mathbf{x}) = \int \eta_{\theta}(y|\mathbf{z}) p_{\theta}(\mathbf{z}|\mathbf{x}) d\mathbf{z}. \quad (14)$$

Similar to (3), we introduce an arbitrary function  $q_{\phi}(\mathbf{z}|\mathbf{x})$  parameterized by  $\phi$

$$\begin{aligned} \hat{\rho}_{\theta}(y|\mathbf{x}) &= \int \eta_{\theta}(y|\mathbf{z}) p_{\theta}(\mathbf{z}|\mathbf{x}) d\mathbf{z} \\ &= E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \left[ \eta_{\theta}(y|\mathbf{z}) \frac{p_{\theta}(\mathbf{z}|\mathbf{x})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right]. \end{aligned}$$

Once again, taking  $-\log$  on both sides leads to

$$-\log \hat{\rho}_{\theta}(y|\mathbf{x}) = -\log E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \left[ \eta_{\theta}(y|\mathbf{z}) \frac{p_{\theta}(\mathbf{z}|\mathbf{x})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right]. \quad (15)$$

Applying Jensen's inequality, we write

$$\begin{aligned} -\log \hat{\rho}_{\theta}(y|\mathbf{x}) &\leq -E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \log \left[ \eta_{\theta}(y|\mathbf{z}) \frac{p_{\theta}(\mathbf{z}|\mathbf{x})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \\ &\leq -E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \log \eta_{\theta}(y|\mathbf{z}) - E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \log \left[ \frac{p_{\theta}(\mathbf{z}|\mathbf{x})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right]. \end{aligned} \quad (16)$$

Following the definition of KL divergence, we write:

$$-\log \hat{\rho}_{\theta}(y|\mathbf{x}) \leq -E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \log \eta_{\theta}(y|\mathbf{z}) + \text{KL}(q_{\mathbf{z}|\mathbf{x}, \phi} \| p_{\mathbf{z}|\mathbf{x}, \theta}). \quad (17)$$

We denote the ground truth class of sample  $\mathbf{x}$  as  $y_{\text{gt}}$ . The true distribution  $\rho$  of  $y$  given  $\mathbf{x}$  can be written as following by using the Kronecker delta function  $\delta[\cdot]$ :

$$\rho(y|\mathbf{x}) = \delta[y - y_{\text{gt}}] = \begin{cases} 1, & y = y_{\text{gt}} \\ 0, & y \neq y_{\text{gt}}. \end{cases} \quad (18)$$

With the above notion, we write the CE between the true and the estimated label distributions as

$$\text{CE}_{y|\mathbf{x}}(\rho, \hat{\rho}_{\theta}) = - \sum_{c \in \mathcal{Y}} \rho(y = c|\mathbf{x}) \log \hat{\rho}_{\theta}(y = c|\mathbf{x}). \quad (19)$$

From (17) and (19), we can derive the variational bound for classification

$$\begin{aligned} \text{CE}_{y|\mathbf{x}}(\rho, \hat{\rho}_{\theta}) &= - \sum_{c \in \mathcal{Y}} \rho(y = c|\mathbf{x}) \log \hat{\rho}_{\theta}(y = c|\mathbf{x}) \\ &\leq - \sum_{c \in \mathcal{Y}} \rho(y = c|\mathbf{x}) \{ E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \log \eta_{\theta}(y = c|\mathbf{z}) \} \\ &\quad + \sum_{c \in \mathcal{Y}} \rho(y = c|\mathbf{x}) \{ \text{KL}(q_{\mathbf{z}|\mathbf{x}, \phi} \| p_{\mathbf{z}|\mathbf{x}, \theta}) \}. \end{aligned} \quad (20)$$

The bound can be further simplified into

$$\begin{aligned} \text{CE}_{y|\mathbf{x}}(\rho, \hat{\rho}_{\theta}) &\leq - \sum_{c \in \mathcal{Y}} \rho(y = c|\mathbf{x}) \{ E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \log \eta_{\theta}(y = c|\mathbf{z}) \} \\ &\quad + \text{KL}(q_{\mathbf{z}|\mathbf{x}, \phi} \| p_{\mathbf{z}|\mathbf{x}, \theta}). \end{aligned} \quad (21)$$

Using (18), we obtain the variational bound for classification

$$\text{CE}_{y|\mathbf{x}}(\rho, \hat{\rho}_{\theta}) \leq -E_{q_{\mathbf{z}|\mathbf{x}, \phi}} \log \eta_{\theta}(y_{\text{gt}}|\mathbf{z}) + \text{KL}(q_{\mathbf{z}|\mathbf{x}, \phi} \| p_{\mathbf{z}|\mathbf{x}, \theta}). \quad (22)$$

Following the approach [38] discussed in Section II-C, we can define the  $\beta$ -VAE loss function for a single sample  $\mathbf{x}$  as:

$$\mathcal{L}_{\theta, \phi}(\mathbf{x}, y_{\text{gt}}) = E_{q_{\mathbf{z}|\mathbf{x}, \phi}} [-\log \eta_{\theta}(y_{\text{gt}}|\mathbf{z})] + \beta E_{q_{\mathbf{z}|\mathbf{x}, \phi}} [-\log p_{\theta}(\mathbf{z}|\mathbf{x})] \quad (23)$$

with a control parameter  $\beta$ . We can interpret the first term as the conditional classification loss and the second term as the coding cost or rate.

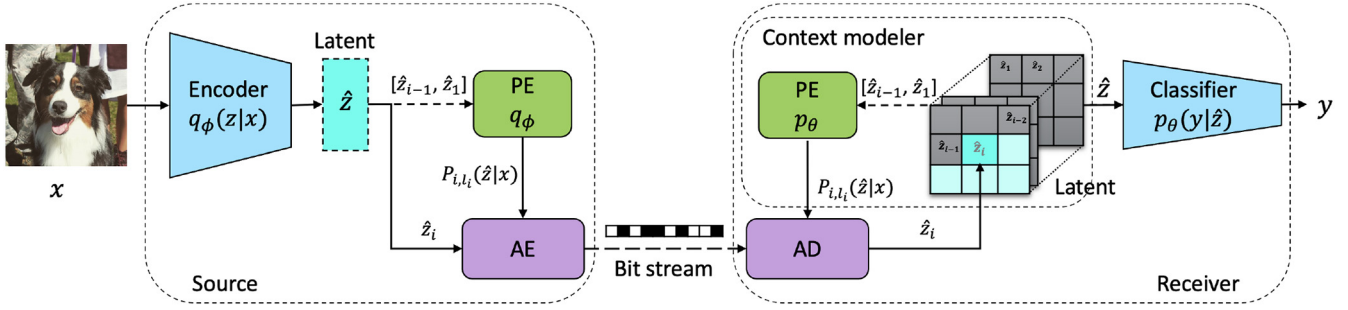


Fig. 3. Overview of the proposed VAE classifier during inference: Quantized latent vector  $\hat{\mathbf{z}}$  is encoded into bit stream by a context-adaptive arithmetic encoder (AE) assisted by probability estimator (PE). At receiver, probability of each symbol  $\hat{z}_i$  (shown in cyan) is estimated by using a learned PE based on previously decoded latents  $\hat{z}_{i-1}, \dots, \hat{z}_1$  (shown in gray). Without groundtruth distribution of the latent elements  $q_\phi(\hat{\mathbf{z}}|\mathbf{x})$  at the receiver, PE learns to approximate  $p_\theta \approx q_\phi$  during training.

### B. Learning Model

Our end-to-end compression-classification learning model consists of an encoder (**E**), a **PE** and a classifier (**CL**) as shown in Fig. 3. Following the approach in [27], we use a context model to estimate the conditional probabilities in **PE**. Consider the following setting:

$$\mathbf{x} \xrightarrow{E} \mathbf{z} \xrightarrow{Q} \hat{\mathbf{z}} \xrightarrow{CL} y \quad (24)$$

$\mathbf{x}$  is the input image to the model with dimensions  $(w \times h \times 3)$ . The encoder, parameterized by  $\phi$ , maps  $\mathbf{x}$  to the latent representation  $\mathbf{z}$  of dimension  $([w/s] \times [h/s] \times K = m)$ . A quantizer (**Q**) further maps  $\mathbf{z}$  to  $\hat{\mathbf{z}}$  by assigning each element  $z_i$  ( $i = 1, \dots, m$ ) to the closest quantization levels  $\mathcal{D} = \{d_\ell, \ell = 1, \dots, L\}$ . For instance, multiple  $z_i$  values can be mapped to the same  $d_\ell$  such that

$$\hat{z}_i = Q(z_i) = \arg \min_{d \in \mathcal{D}} \|z_i - d\|_2. \quad (25)$$

The quantization centers  $\{d_\ell\}$  are learned through training. The **CL**, parameterized by  $\theta$ , takes the  $\hat{\mathbf{z}}$  as the input and predicts the class label  $y$  for the given image sample  $\mathbf{x}$ . Taking quantization into account for practical applications, (23) can be rewritten as follows:

$$\mathcal{L}_{\theta, \phi}(\mathbf{x}, y_{\text{gt}}) = E_{q_{\mathbf{z}|\mathbf{x}, \phi}}[-\log \eta_\theta(y_{\text{gt}}|\hat{\mathbf{z}})] + \beta E_{q_{\hat{\mathbf{z}}|\mathbf{x}, \phi}}[-\log p_\theta(\hat{\mathbf{z}}|\mathbf{x})]. \quad (26)$$

We now examine how to obtain each term for the loss function in (26).

### C. Rate Loss

The second RHS term of (26) represents the CE between two conditional distributions of  $\mathbf{z}$ :  $p_\theta(\hat{\mathbf{z}}|\mathbf{x})$  and  $q_\phi(\hat{\mathbf{z}}|\mathbf{x})$ . The distribution  $q_\phi(\hat{\mathbf{z}}|\mathbf{x})$  is deterministic and readily available after the deterministic encoding step (**E**). During training, we estimate the distribution  $p_\theta(\hat{\mathbf{z}}|\mathbf{x})$  with a **PE** using a conditional context model following [27], [38]. The latent representation  $\hat{\mathbf{z}}$  as discussed in Section III-B is a 3-D tensor containing  $m$  number of latent elements. We can index this 3-D tensor  $\hat{\mathbf{z}}$  given  $\mathbf{x}$  as a vector in raster-scan order and write  $p_\theta(\hat{\mathbf{z}}|\mathbf{x})$  as

$$p_\theta(\hat{\mathbf{z}}|\mathbf{x}) = \prod_{i=1}^m p_\theta(\hat{z}_{i-1}, \dots, \hat{z}_1, \mathbf{x}). \quad (27)$$

With **PE**, we can efficiently estimate the conditional probability of the latent value  $\hat{z}_i$  that takes the value of the quantization center  $d_{\ell_i}$  which can be defined as  $P_{i, \ell_i}(\hat{\mathbf{z}}|\mathbf{x}) = p_\theta(\hat{z}_i = d_{\ell_i} | \hat{z}_{i-1}, \dots, \hat{z}_1, \mathbf{x})$  for notional simplicity. With this notation, can write the approximate coding rate ( $\mathcal{L}_R$ ) as follows:

$$\begin{aligned} \mathcal{L}_{R(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x}) &= \text{CE}_{\hat{\mathbf{z}}|\mathbf{x}}(q_\phi, p_\theta) \\ &= E_{q_{\hat{\mathbf{z}}|\mathbf{x}, \phi}}[-\log p_\theta(\hat{\mathbf{z}}|\mathbf{x})] \\ &= -\sum_{i=1}^m \sum_{\ell_i=1}^L q_\phi(\hat{z}_{i, \ell_i}|\mathbf{x}) \log P_{i, \ell_i}(\hat{\mathbf{z}}|\mathbf{x}). \end{aligned} \quad (28)$$

Here,  $q_\phi(\hat{z}_{i, \ell_i}|\mathbf{x})$  is the probability of  $\Pr(\hat{z}_i = d_{\ell_i}|\mathbf{x})$ . Any  $\hat{z}_i$  can take only one particular quantization value  $d_{\ell_i}$ .

Similar to [27], through end-to-end training, we learn the distribution  $p_\theta(\hat{\mathbf{z}}|\mathbf{x})$  in order to optimally approximate  $p_\theta \approx q_\phi$ . Through context-based adaptive arithmetic encoding process (CABAC) [45],  $q_\phi(\hat{\mathbf{z}}|\mathbf{x})$  is available for the probability estimation, only at the encoder. Once the encoded bit-stream is received at the context-based adaptive arithmetic decoder, the learned conditional probability  $p_\theta(\hat{\mathbf{z}}|\mathbf{x})$  is used for probability estimation to recover  $\hat{\mathbf{z}}$  whereas  $q_\phi(\hat{\mathbf{z}}|\mathbf{x})$  is not available [46]. For this reason, we learn the distribution  $p_\theta(\hat{\mathbf{z}}|\mathbf{x})$  to approximate  $p_\theta \approx q_\phi$  which makes  $p_\theta$  the “context modeler” as commonly referred in [45] and [46]. See Fig. 3. When the learning succeeds by training, with (28), we observe that the coding rate  $\mathcal{L}_{R(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x})$  reduces to the entropy  $H(\hat{\mathbf{z}}|\mathbf{x})$ .

### D. Classification Loss

We interpret the first RHS term of (26) as the classification loss  $\mathcal{L}_{\text{CL}(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x}, y_{\text{gt}})$ . Following the rate approach as given in Section III-C, we can rewrite the classification loss term as follows:

$$\begin{aligned} \mathcal{L}_{\text{CL}(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x}, y_{\text{gt}}) &= E_{q_{\hat{\mathbf{z}}|\mathbf{x}, \phi}}[-\log \eta_\theta(y_{\text{gt}}|\hat{\mathbf{z}})] \\ &= -\sum_{i=1}^m \sum_{\ell_i=1}^L q_\phi(\hat{z}_{i, \ell_i}|\mathbf{x}) \log Q_{i, \ell_i}(y_{\text{gt}}|\hat{\mathbf{z}}). \end{aligned}$$

Here, we have to estimate the class label probability for each  $\hat{z}_{i, \ell_i}$  which is defined as  $Q_{i, \ell_i}(y_{\text{gt}}|\hat{\mathbf{z}}) = \eta_\theta(y_{\text{gt}}|\hat{z}_i = d_{\ell_i})$ . We note that estimating the conditional label probability given each individual  $\hat{z}_i$  can be challenging.

On the other hand, given  $\hat{\mathbf{z}}$ , we can efficiently predict the class label using a trained CNN classifier such as ResNet [47]. Hence, we can easily estimate  $-\log \eta_{\theta}(y_{\text{gt}}|\hat{\mathbf{z}})$  given the entire sequence of  $\hat{\mathbf{z}}$  rather than for one individual element  $\hat{z}_i$  therein. In standard stochastic gradient optimizers, we have only one sequence sample  $\hat{\mathbf{z}}$  at a time. Hence, the stochastic expectation of  $E_{q_{\hat{\mathbf{z}}|\mathbf{x},\phi}}[-\log \eta_{\theta}(y_{\text{gt}}|\hat{\mathbf{z}})]$  can be approximated simply by  $-\log \eta_{\theta}(y_{\text{gt}}|\hat{\mathbf{z}})$  [37]. We can reduce this approximation error by iterating sufficiently over same  $\mathbf{x}$ . This simple approximation leads to general classification loss

$$\begin{aligned}\mathcal{L}_{\text{CL}(\theta,\phi)}(\hat{\mathbf{z}}|\mathbf{x}, y_{\text{gt}}) &= E_{q_{\hat{\mathbf{z}}|\mathbf{x},\phi}}[-\log \eta_{\theta}(y_{\text{gt}}|\hat{\mathbf{z}})] \\ &\approx -\log \eta_{\theta}(y_{\text{gt}}|\hat{\mathbf{z}}).\end{aligned}\quad (29)$$

With this formulation, we express the total loss  $\mathcal{L}_{\theta,\phi}(\mathbf{x}, y_{\text{gt}})$  of the joint classification and compression model for image sample  $\mathbf{x}$  as

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(\mathbf{x}, y_{\text{gt}}) &= \mathcal{L}_{\text{CL}(\theta,\phi)}(\hat{\mathbf{z}}|\mathbf{x}, y_{\text{gt}}) + \beta \mathcal{L}_{\text{R}(\theta,\phi)}(\hat{\mathbf{z}}|\mathbf{x}) \\ &\approx -\log \eta_{\theta}(y_{\text{gt}}|\hat{\mathbf{z}}) \\ &\quad - \beta \sum_{i=1}^m \sum_{\ell_i=1}^L q_{\phi}(\hat{z}_{i,\ell_i}|\mathbf{x}) \log P_{i,\ell_i}(\hat{\mathbf{z}}|\mathbf{x}).\end{aligned}\quad (30)$$

Similar to the rate-distortion loss in reconstruction [27], [38], [48],  $\beta$  controls the rate versus mis-classification tradeoff.

#### IV. EXPERIMENTS AND RESULTS

This section presents several experiment results of classification using the proposed VAE approach of Section III. We tested the accuracy of classification on three well-known data sets: CIFAR-10 [49], CIFAR-100 [50], and ImageNet-1k [51]. CIFAR-10 consists of ten classes with 5000 images for training and 1000 images for testing per class. CIFAR-10 images are in RGB format of size  $32 \times 32$ . Similar to CIFAR-10, CIFAR-100 data set contains 50k training and 10k test RGB images of size  $32 \times 32$  in 100 classes. ImageNet-1k consists of 1000 classes, each of which consists of up to 1300 training images and 50 validation images of mostly  $256 \times 256$  RGB.

Our experiments used the same quantizer and the context model as [27] to estimate probabilities. Instead of separately training the encoder-decoder and the context model (PE) as in [27], the training of PE takes place simultaneously with encoder and classifier training.

##### A. VAE-Based Joint Compression and Classification Models

When designing the joint model, we began with a ResNet [47] and split into two parts as Encoder (E) and classifier (CL). Next, we added a batch normalization layer and a quantizer block to the encoder and further modified the number of filters and strides. For each CIFAR-10, CIFAR-100, and ImageNet-1k data sets, we considered two different encoders-classifier combinations. Fig. 5(a)–(g) depict the structural details of four proposed VAE models labeled as AE-V<sub>1</sub>, AE-V<sub>2</sub>, AE-V<sub>3</sub>, and AE-V<sub>4</sub>. We use typical notations in CNNs to denote their components. For example, “Conv 16 ( $3 \times 3$ )” represents a 2-D convolution block with 32 filters of size  $3 \times 3$  and a stride of 2. “Res 16 ( $3 \times 3$ )” denotes a basic ResNet block with a down-sampling factor of 2.

To compare the performance gains of the proposed approach, we mainly used JPEG (4:2:0)<sup>1</sup> as the benchmark technique to compress images within the data sets at different quality ( $Q$ ) values. One can vary the quality value  $Q \in [1, 100]$  during compression to set image quality. We then trained a ResNet classifier based on these compressed images for each  $Q$  value. To calculate the required channel bandwidth (i.e., image size), we averaged the bits-per-pixel (bpp) for each coded image over the data set. For a fair comparison, we only consider the data bits of the images according to bpp without counting packet headers. Following similar steps, we generated rate-accuracy performance for popular standard codecs: JPEG2000,<sup>2</sup> WebP,<sup>3</sup> and BPG,<sup>4</sup> to be used as baselines.

In the joint compression and classification model, we vary the number of quantization centers ( $L$ ) to compress images at different BPP values without changing the E and CL model architectures. As proposed in [27], one can alter the encoder-decoder/classifier architecture to generate latent of different dimensions to obtain the requisite bpp as well.

We selected  $\beta = 2$  for all experiments. During model training, we used a “Momentum” Optimizer for the E-CL part of the network and an “Adam” optimizer for the PE with a learning rate of 0.0005 and reduced the learning rate by  $\times 0.1$  at 0.25 and 0.75 of total epochs over 90 epochs. In contrast to [27], we concurrently train both E-CL and PE to also enable back-propagation from PE to E. This configuration facilitates model convergence at lower bpp.

##### B. CIFAR-10 Experiments

For CIFAR-10, we used the models AE-V<sub>1</sub> and AE-V<sub>2</sub> of different complexities. Fig. 5(a)–(d) provide the structural details of both VAE models. As the baseline classification model, we used a ResNet-18. Fig. 4(a) shows the result for CIFAR-10 data set with different combinations of E-CL in comparison with popular standard codecs: JPEG, JPEG2000, WebP, and BPG. Both AE-V<sub>1</sub> and AE-V<sub>2</sub> achieve substantial performance improvement over the traditional JPEG (4:2:0) in terms of rate (bpp) and accuracy tradeoff. At 0.8 bit/pixel, our joint compression-classification accuracy improves from 77% to 87%. More importantly, this performance improvement is achieved with significantly lighter-weight models, resulting impressive inference speed improvements and power savings compared to the baseline classifiers as will be discussed in Section V. Further, compared to JPEG2000, WebP, and BPG, the proposed VAE-based codecs maintain the rate-accuracy performance specially at lower bandwidth end. In particular, both VAE-based models demonstrate over 5% accuracy improvement at 0.9 bpp compared to BPG.

##### C. CIFAR-100 Experiments

As the baseline classification model, we used a ResNet-18 model with a fully connected layer of 100 neurons

<sup>1</sup><http://www.openjpeg.org/>

<sup>2</sup><https://kakadusoftware.com/>

<sup>3</sup><https://developers.google.com/speed/webp/download>

<sup>4</sup><https://bellard.org/bpg/>

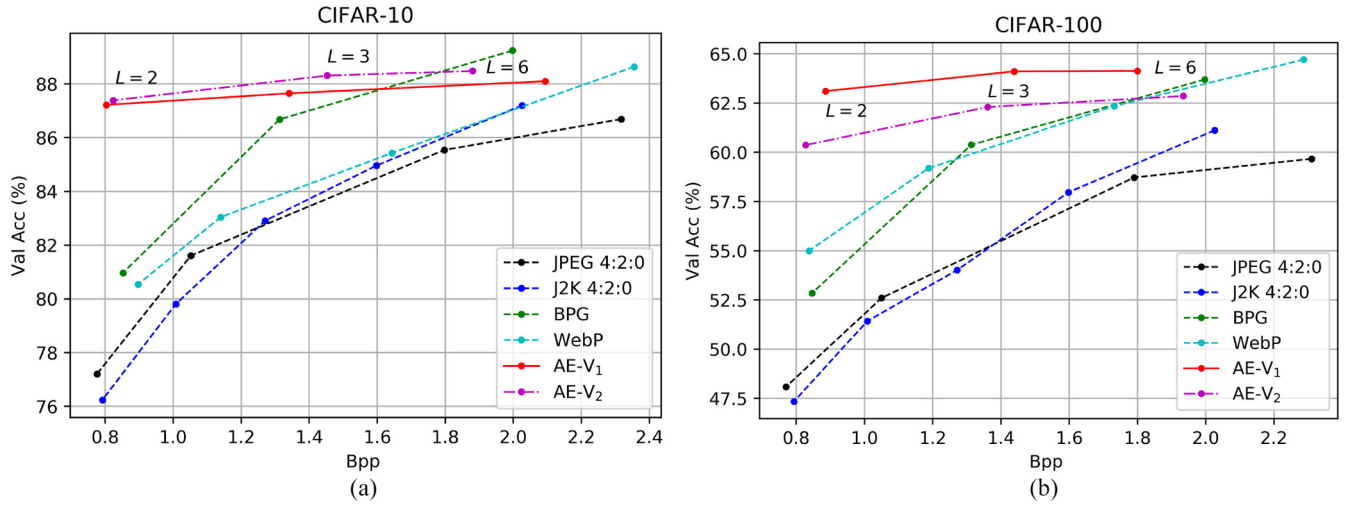


Fig. 4. Classification accuracy versus rate results for end-to-end compression and classification on (a) CIFAR-10 and (b) CIFAR-100 data sets. The proposed VAE-based compression and classification framework outperforms popular commercial image compression codecs in terms of rate-accuracy, at lower bandwidths.

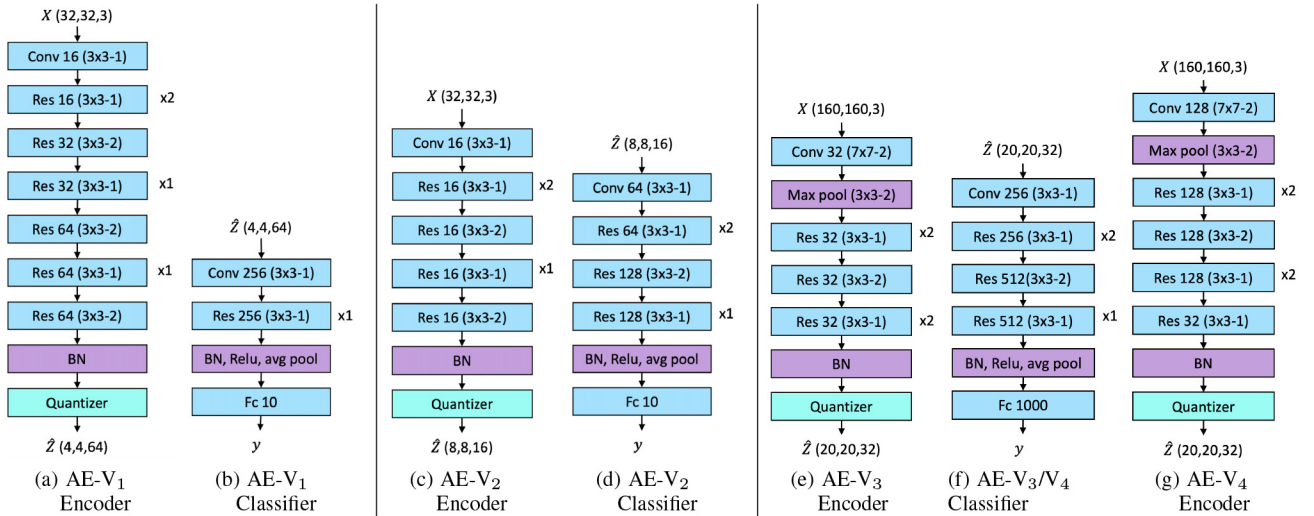


Fig. 5. Encoders and classifiers for the proposed end-to-end compression and classification models AE-V<sub>1</sub>, AE-V<sub>2</sub>, AE-V<sub>3</sub>, and AE-V<sub>4</sub>. “Conv 16 ( $3 \times 3-2$ )” represents a 2-D convolution block with 32 filters of size  $3 \times 3$  and stride of 2. “Res 16 ( $3 \times 3-2$ )” represents a basic ResNet block [47] with down-sampling factor 2. (a) AE-V<sub>1</sub> encoder. (b) AE-V<sub>1</sub> classifier. (c) AE-V<sub>2</sub> encoder. (d) AE-V<sub>2</sub> classifier. (e) AE-V<sub>3</sub> encoder. (f) AE-V<sub>3</sub>/V<sub>4</sub> classifier. (g) AE-V<sub>4</sub> encoder.

instead of 10 in the original model designed for CIFAR-10. For the VAE models for classification, we used AE-V<sub>1</sub> and AE-V<sub>2</sub>, each with same encoder [shown in Fig. 5(a) and (c)] and the Classifier (Fig. 5(b) and (d)) with a fully connected layer of 100 neurons (Fc-100) instead of Fc-10. For training, we follow the same settings as in CIFAR-10. Fig. 4(b) compared the rate-accuracy performance of AE-V<sub>1</sub> and AE-V<sub>2</sub> models for CIFAR-100 validation set. The proposed method achieves similar improvement as observed in CIFAR-10.

#### D. ImageNet Experiments

For ImageNet experiments, we followed the same process as used for CIFAR-10 to generate the JPEG (4:2:0) baseline. As a part of data augmentation, we resized the images to  $256 \times 256$  and cropped them down to  $160 \times 160$  in training and testing to obtain the desired dimensions of the latent  $\hat{z}$ . We propose

modified VAE models AE-V<sub>3</sub> and AE-V<sub>4</sub> for ImageNet. See Fig. 5(e)–(g) for model architectures.

For a fair performance comparison, we fine-tuned the baseline ResNet models with the JPEG encoded images at each quality level ( $Q$ ) to record the best accuracy achievable at given  $Q$  as benchmark results. For the same reason, we do not include packet header in bpp calculation.

Testing the E-CL combinations AE-V<sub>3</sub> and AE-V<sub>4</sub> on ImageNet generates the results of Fig. 6. The performance of the two shallower VAEs is compared against the benchmark ResNet-18 and ResNet-34 operated on JPEG encoding. Our results demonstrate clear performance improvement in terms of rate-accuracy tradeoff, without increasing complexity as discussed in Section V. In fact, the proposed VAE delivers comparable performance as conventional ResNet-34 that requires 21.8M parameters which is twice as many compared to AE-V<sub>3</sub> and AE-V<sub>4</sub> models.



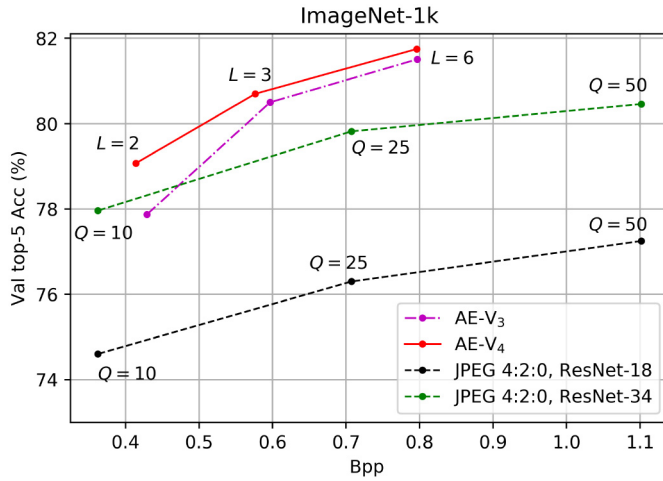


Fig. 6. Classification accuracy versus rate on ImageNet-1k for end-to-end compression and classification. The proposed VAE-based compression and classification framework (AE-V<sub>4</sub>) significantly outperforms JPEG commercial image compression codes in terms of rate-accuracy.

TABLE II  
COMPLEXITY COMPARISON OF THE MODELS FOR CIFAR-10 AND IMAGENET. ALL PARAMETERS ARE IN MILLIONS (M). THE PERCENTAGE REDUCTION IS CALCULATED COMPARED TO THE BASELINE

Data set	Model	Encoder	Classifier	Baseline
CIFAR-10	AE-V <sub>1</sub>	0.245 (65%↓)	0.372 (47%↓)	0.704
	AE-V <sub>2</sub>	0.025 (96%↓)	0.687 (2%↓)	ResNet-18
ImageNet-1k	AE-V <sub>3</sub>	0.100 (99%↓)	11.35 (3%↓)	11.7
	AE-V <sub>4</sub>	1.550 (87%↓)	11.35 (3%↓)	ResNet-18
ImageNet-1k	AE-V <sub>3</sub>	0.100 (99%↓)	11.35 (48%↓)	21.8
	AE-V <sub>4</sub>	1.550 (99%↓)	11.35 (48%↓)	ResNet-34

## V. COMPLEXITY COMPARISON

Simpler encoder models are practically more favorable for wide deployment on low cost devices, which are severely constrained in memory, power, and computation capacity. In this section, we compare the complexity of the proposed models in terms of model size (number of parameters), inference speed, and power consumption to the baseline models.

### A. Model Size Comparison

Considering practical constraints in low-cost sensors, our encoder design has significantly fewer number of parameters in comparison with the typical cloud-based classifier. See Table II for a complexity comparison in terms of the number of parameters for the VAE models used in CIFAR-10 and ImageNet-1k experiments. Percentages provided in the table indicates the reduction of the number of parameters compared to the baseline classifier.

All the proposed encoder and classifier models show parameters savings compared to their corresponding baseline models. The achieved parameter savings range from 65%–99% for encoders and 2%–48% for classifiers. Such balance of complexity reductions are highly practical for wide IoT deployment since classifiers at the server side are equipped with more

TABLE III  
SPEED COMPARISON IN TERMS OF “IPS.” THE INFERENCE RESULTS ARE BASED ON A NVIDIA TITAN-V GPU. THE PROPOSED VAE-BASED CLASSIFIER IS  $\times 1.5$  AND  $\times 2.25$  FASTER COMPARED TO RESNET-18 AND RESNET-34 BASELINES

Model	Encode (E)	Classifier (CL)	E-CL	Baseline CL
AE-V <sub>1</sub>	1702	20283 ( $\times 13.1\uparrow$ )	1570( $\uparrow$ )	1547
AE-V <sub>2</sub>	2114	6524 ( $\times 4.2\uparrow$ )	1596( $\uparrow$ )	ResNet-18
AE-V <sub>3</sub>	4691	5122 ( $\times 1.5\uparrow$ )	2448( $\downarrow$ )	3330
AE-V <sub>4</sub>	2417		1642( $\downarrow$ )	ResNet-18
AE-V <sub>3</sub>	4691	5122 ( $\times 2.25\uparrow$ )	2448( $\uparrow$ )	2268
AE-V <sub>4</sub>	2417		1642( $\downarrow$ )	ResNet-34

resources compared to the source encoders at the embedded devices.

1) *Task Versus Model Size*: When comparing the model complexity versus task difficulty, more challenging classification tasks often require more complex encoders. Note that both models AE-V<sub>3</sub> and AE-V<sub>4</sub> use the same network architecture as the classifier for ImageNet data set. Furthermore, from Table II, it is clear that the proposed VAE classifiers of AE-V<sub>3</sub> and AE-V<sub>4</sub> have the same complexity in terms of the number of parameters while AE-V<sub>4</sub> model uses a relatively more complex encoder than AE-V<sub>3</sub>. Since complex encoders with more parameters are capable of extracting more discriminate features, Fig. 6 shows AE-V<sub>4</sub> with higher rate-classification performance than AE-V<sub>3</sub>, especially when handling more challenging tasks such as ImageNet classification with 1000 classes.

We observe similar trend in rate classification in Fig. 4(b) for CIFAR-100 data set with 100 classes: A relatively increase of encoder complexity may achieve better performance than a simple encoder through feature extraction for classification. Note in Table II that the model complexity in number of parameters is  $\times 10$  smaller for AE-V<sub>2</sub> when compared to AE-V<sub>1</sub>. However, for a relatively simple classification task such as CIFAR-10 with only ten classes, a simpler encoder used in AE-V<sub>2</sub> is stronger to preserve enough features to generate the latent vector. In fact, even in such case, having a relatively complex classifier with more parameters to process the extracted features has some benefit, although the resulting rate-classification performance improvement is rather modest.

### B. Inference Speed Comparison

We also observe considerable reduction in inference time by directly classifying on the latent/feature maps without reconstruction. In Table III, we list the inference speed in terms of average images per second (ips) for CIFAR-10 test set and ImageNet-1k validation set.

The proposed end-to-end-trained AE-V<sub>1</sub> and AE-V<sub>2</sub> classifiers record  $\times 13.1$  and  $\times 4.2$  speed gains on CIFAR-10 data set, respectively, compared to the ResNet-18 classifier. Image patches used at inference have the size of  $32 \times 32$ . Similarly, our proposed classifier is  $\times 1.5$  faster in comparison with the ResNet-18 baseline and  $\times 2.25$  faster in comparison with ResNet-34 baseline, on ImageNet-1k data set where image

TABLE IV  
NUMBER OF FLOPs ( $\times 10^9$ ) COMPARISON FOR THE PROPOSED MODELS

Model	Encoder	Classifier	Baseline
Input: $32 \times 32 \times 3$			
AE- $V_1$	0.037 ( <b>93%</b> ↓)	0.005 ( <b>99%</b> ↓)	0.557
AE- $V_2$	0.013 ( <b>97%</b> ↓)	0.013 ( <b>97%</b> ↓)	ResNet-18
Input: $160 \times 160 \times 3$			
AE- $V_3$	0.107 ( <b>88%</b> ↓)	1.229 ( <b>32%</b> ↑)	0.928
AE- $V_4$	1.399 ( <b>50%</b> ↑)	1.229 ( <b>32%</b> ↑)	ResNet-18
Input: $160 \times 160 \times 3$			
AE- $V_3$	0.107 ( <b>94%</b> ↓)	1.229 ( <b>34%</b> ↓)	1.873
AE- $V_4$	1.399 ( <b>25%</b> ↓)	1.229 ( <b>34%</b> ↓)	ResNet-34

patches have the size of  $160 \times 160$ . Since the inference speed of the encoder for the baseline can vary based on the codec (JPEG, BPG etc.), we only highlight the inference speed gains of the classifiers (CL) for fairness. Moreover, due to the simplicity of the proposed AE- $V_1$  and AE- $V_2$  encoders, the inference speed including the encoding time (E-CL column) surpasses the baseline CL speed calculated even without the encoding time.

### C. Power Savings Comparison

When measured on the same device (GPU), the number of floating point operations per second (FLOPS) is directly proportional to the power consumption. The proposed encoders are not only less complex in terms of a number of parameters, but also demand significantly lower number of arithmetic operations which is an essential feature for IoT, mobile and wireless applications [52]. In Table IV we list the required computational operations in FLOPs for the proposed encoders. We indicate the power savings as a percentage compared to the baseline.

The proposed encoders and classifiers of AE- $V_1$  and AE- $V_2$  models achieve power savings in the range of 93%–97% compared to ResNet-18 classifier. AE- $V_3$  and AE- $V_4$  models with larger input image sizes show less power savings in the classifier compared to the encoder due to the high number of parameters. However, this low power savings are compensated by the resource rich servers at the cloud. More importantly, the proposed encoders display significant power savings that enable wide deployment of power-constrained low-end source embedded devices. Modern smart phones are capable of providing over  $10 \times 10^9$  FLOPs [52], [53].

## VI. DISCUSSION

### A. Joint Compression and Classification With Simultaneous Reconstruction

Thus far, our proposed joint compression and classification model allows the classifier to operate directly in latent space. We bypass the image reconstruction step unlike the image/video systems [13]–[15], [20], [41] that perform classification on reconstructed data. In practical systems, however, there also exist several application scenarios which may require both autonomous image classification and image reconstruction for users. For example, RGB images may need

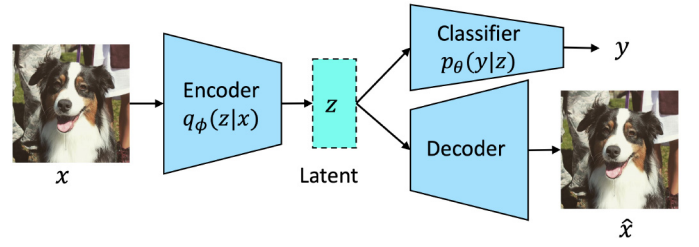


Fig. 7. Proposed joint classification and compression model with reconstruction. Encoder transforms input  $\mathbf{x}$  to a latent vector  $\mathbf{z}$  that is optimized for rate-classification-distortion performance.

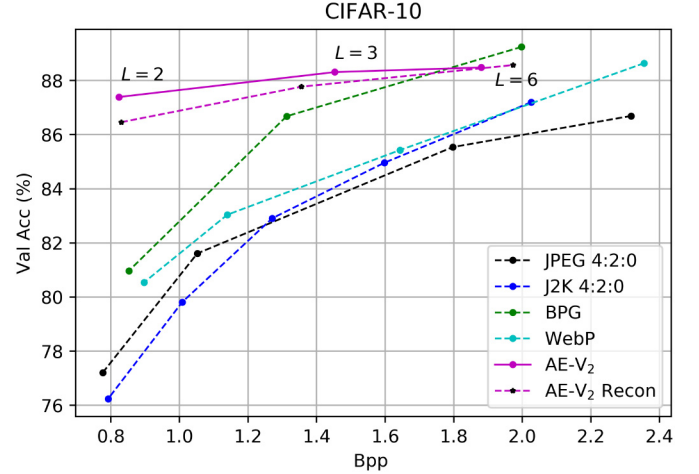


Fig. 8. Classification accuracy versus rate on CIFAR-10 for joint compression-classification with reconstruction: AE- $V_2$ -Recon. Note that rate-accuracy performance of AE- $V_2$ -Recon is still sufficiently good compared to popular commercial codecs, at lower rates.

to be stored on the cloud to be retrieved later after successful classification. These reconstructed and stored RGB images may be required for other subsequent learning tasks, such as object detection and segmentation trained on RGB images. To accommodate such potential dual compression objectives of simultaneous classification and reconstruction, we further modify the current VAE to incorporate a parallel decoder at the remote node (or cloud).

As shown in Fig. 7, we include a basic DL-based image decoder (D) to the proposed joint compression and classification model. This framework of joint compression and classification model with reconstruction (E-CL-D) can be trained similarly to the proposed VAE. In consideration of the reconstruction accuracy, we construct a modified loss function of

$$\mathcal{L} = \mathcal{L}_C + \beta \mathcal{L}_R + \gamma \mathcal{L}_D. \quad (31)$$

We use the mean square error (MSE) between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  as the distortion loss  $\mathcal{L}_D$  and empirically choose a control parameter  $\gamma$  smaller than  $\beta$  to favor high classification accuracy over rate-distortion.

In our experiment, we applied a DL decoder similar to the architecture of [27] with modified numbers of convolution kernels to reduce the number of parameters. The DNN decoder has 0.308 M parameters which is only 44% of the ResNet-18 baseline model. We fine-tuned the training of the proposed joint architecture with the modified loss. Fig. 8 shows the

TABLE V  
RECONSTRUCTION QUALITY FOR CIFAR-10 AND IMAGENET-1K.  
L IS THE NUMBER OF QUANTIZATION CENTERS

Data set	$L=2$	$L=3$	$L=6$
	PSNR (dB)		
CIFAR-10	19.24	21.42	22.84
ImageNet-1k	20.18	21.68	22.22
	MS-SSIM		
ImageNet-1k	81.04	86.73	88.35

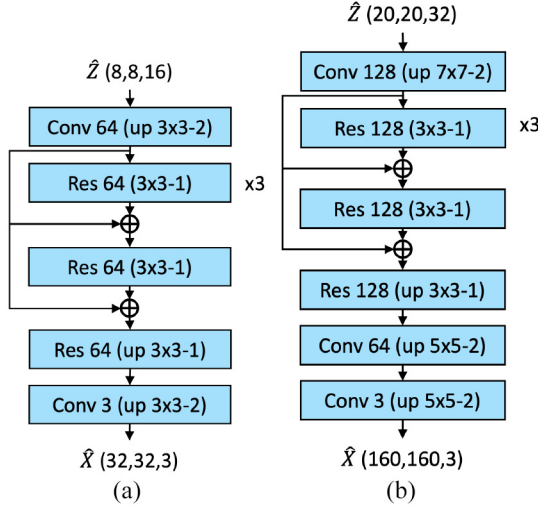


Fig. 9. Decoder design for AE-V<sub>2</sub> and AE-V<sub>3</sub>. “up” indicates up-sampling with transposed convolutions. (a) AE-V<sub>2</sub> decoder. (b) AE-V<sub>3</sub> decoder.

classification accuracy for **E-CL-D** model for CIFAR-10. Note that the classification accuracy loss due to reconstruction is smaller compared to the rate-accuracy gain from standard JPEG, JPEG2000, WebP, and BPG codecs.

For joint compression classification and reconstruction experiments on ImageNet-1k, we would freeze the Encoder-Classifer (**E-CL**) and **PE** parameters optimized for joint compression and classification as described in (30) for AE-V<sub>3</sub> model and retrained only the decoder to minimize the MSE between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$ . A frozen **E-CL** and **PE** guarantees the same rate-classification performance as in Fig. 6 with reconstruct ability. Table V lists reconstruction PSNR and MS-SSIM values for the two test data sets at three bpp settings. The observed lower image quality, observed for the images reconstructed from the rate-accuracy optimized latents, is consistent with the reconstruction quality recorded in [20] for COCO-2017 [54] data set when the codec is optimized for joint compression and object detection.

### B. Visualization of Reconstructed Images

As the decoder models, we used modified architectures of the decoders proposed in [27]. See Fig. 9. “Up” indicated the up-sampling with transposed convolution. Fig. 10(a) and (b) show some of the reconstructed images from “airplane” class of CIFAR-10 [49] and ImageNet-1k [51]. The reconstructed results look blurry compared to the original images since the MSE loss is minimized by the average of the image. We

observe significantly higher distortion when only two quantization levels are used during compression even though the classification accuracy is less compromised (in Fig. 8) as required.

### C. Robustness to Visual Corruptions

In this section, we demonstrate the robustness of the proposed model against common visual corruptions of images based on CIFAR-10-C [55] data set. CIFAR-10-C data set provides corrupted versions of CIFAR-10 [49] test data set for visual corruptions such as noise, blur, pixelation, etc., at five different severity levels of each corruption.

We selected the proposed AE-V<sub>2</sub> VAE model with  $L = 6$  trained on CIFAR-10 data as discussed in Section IV-A during inference on CIFAR-10-C test data set. For nine types of corruptions, namely, *Fog*, *motion blur*, *defocus blur*, *frost*, *pixelation*, *elastic transformation*, *impulse noise*, *Gaussian noise*, and *shot noise*, we repeated the above inference for classification at five severity levels. Fig. 11(a)–(c) show the results. “RGB” baseline corresponds to the inference accuracy with ResNet-20 classifier trained on CIFAR-10 training set.

We note that AE-V<sub>2</sub> VAE classifier outperforms ResNet-20 baseline classifier at inference on images corrupted with fog, motion blur and defocus blur at 3–5 severity levels by a significant margin. Further the proposed model performs slightly worse than the baseline classifier on images corrupted with impulse, Gaussian, and shot noise.

### D. Visualization of Latent Maps

Fig. 12(a)–(c) show the latent maps  $\hat{\mathbf{z}}$  of a sample image  $\mathbf{x}$  for  $L = 6, 3, 2$  related to the model AE-V<sub>3</sub>. For each sub figure, top-left color image is the original and bottom left color image is the reconstructed from the latent maps visualized next to them. We visualize 32 latent maps (each of size  $20 \times 20$ ) before ( $\mathbf{z}$ ) and after quantization maps ( $\hat{\mathbf{z}}$ ). Starting from the first row, the  $\mathbf{z}$  maps are given in every other row. The corresponding quantized  $\hat{\mathbf{z}}$  is given right below each  $\mathbf{z}$  map. Each map is normalized before visualization.

Observe that for each number of quantization centers  $L$ , the quantized maps  $\hat{\mathbf{z}}$  has only  $L$  different colors. Some of the  $\hat{\mathbf{z}}$  maps for  $L = 6$  show clearly identifiable snow leopard figures implying that high level information of  $\mathbf{x}$  is preserved during training and helpful for classification.

### E. Implementation Details: Effect of $\beta$ Adjustment

Recall that the VAE loss of the proposed method for a sample  $\mathbf{x}$  is given by

$$\mathcal{L}_{\theta, \phi}(\mathbf{x}, y_{\text{gt}}) = \mathcal{L}_{C(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x}, y_{\text{gt}}) + \beta \mathcal{L}_{R(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x}) \quad (32)$$

where  $\mathcal{L}_{R(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x})$  is the bandwidth of  $\mathbf{z}$  transmitted over a network link. We measure the bandwidth (rate) in BPP in our experiments. The effect of the tradeoff parameters  $\beta$  for CIFAR-10 data set is shown in Fig. 13(a).

In order to achieve optimal rate-classification accuracy performance at a given rate ( $r_t$ ), we minimize the following



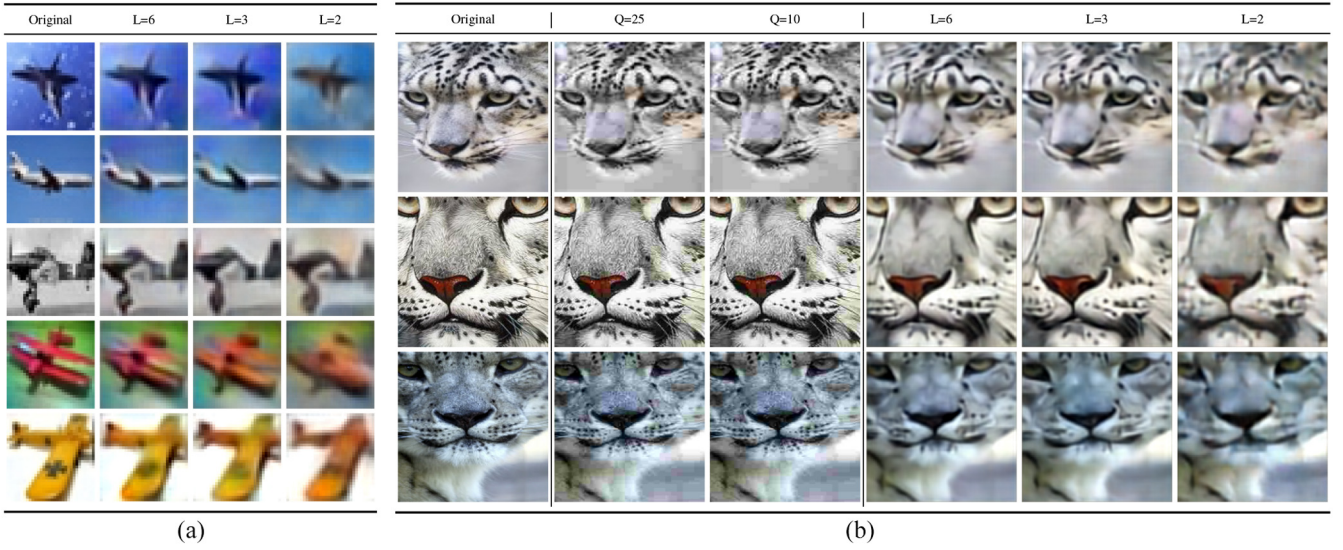


Fig. 10. Examples of (a) CIFAR-10 and (b) ImageNet-1k reconstructed images from the latent space at  $L = 6, 3$ , and  $2$ . In (b),  $Q = 25$  and  $Q = 10$  show JPEG compressed images. We observe significantly higher distortion when only two quantization levels are used during compression.

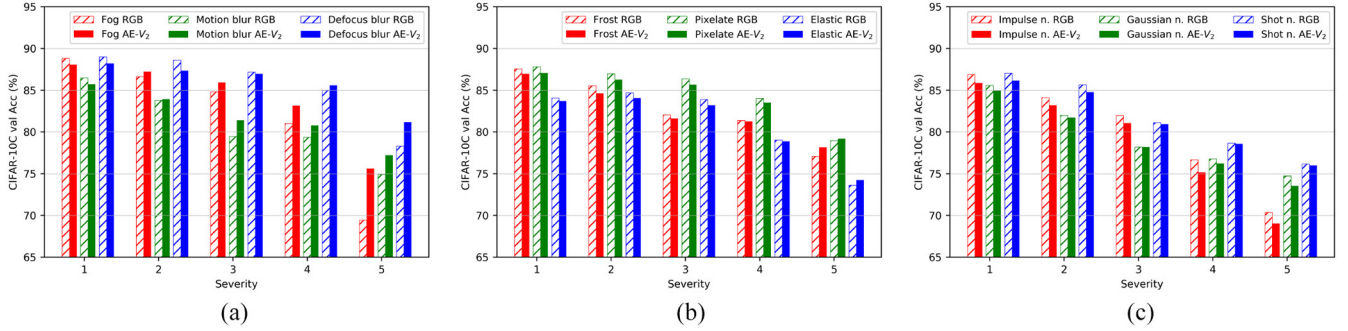


Fig. 11. Robustness comparison for visual corruptions: fog, motion blur, defocus blur, frost, pixelation, elastic transformation, impulse noise, Gaussian noise, and shot noise on CIFAR-10-C data set. (a) Fog, motion blur, and defocus blur. (b) Frost, pixelation, and elastic transformation. (c) Impulse noise, Gaussian noise, and shot noise.

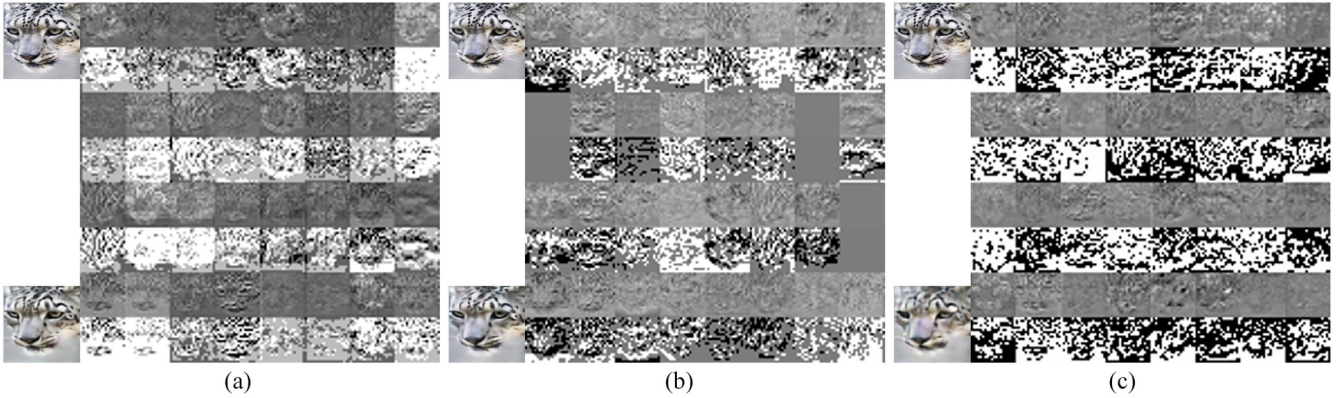


Fig. 12. Latent maps  $\hat{z}$  of a sample image  $x$  for  $L = 6, 3$ , and  $2$  related to the model AE-V<sub>3</sub>. For each sub figure, a top-left color image is the original and bottom-left color image is the reconstructed from the latent maps visualized next to them. Some of the  $\hat{z}$  maps for  $L = 6$  show clearly identifiable snow leopard figures implying that high-level information of  $x$  is preserved during training and helpful for classification. (a)  $L = 6$ . (b)  $L = 3$ . (c)  $L = 2$ .

loss instead of the loss given in (32):

$$\mathcal{L}_{\theta, \phi}(\mathbf{x}, y_{\text{gt}}) = \mathcal{L}_{\text{C}(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x}, y_{\text{gt}}) + \beta \max(\mathcal{L}_{\text{R}(\theta, \phi)}(\hat{\mathbf{z}}|\mathbf{x}) - r_t, 0). \quad (33)$$

We observed better classification accuracy by this approach at a rate slightly above the requirement  $r_t$ . In Fig. 13(b), we illustrate the loss curves for a selected  $\beta$  and an  $r_t$  value.



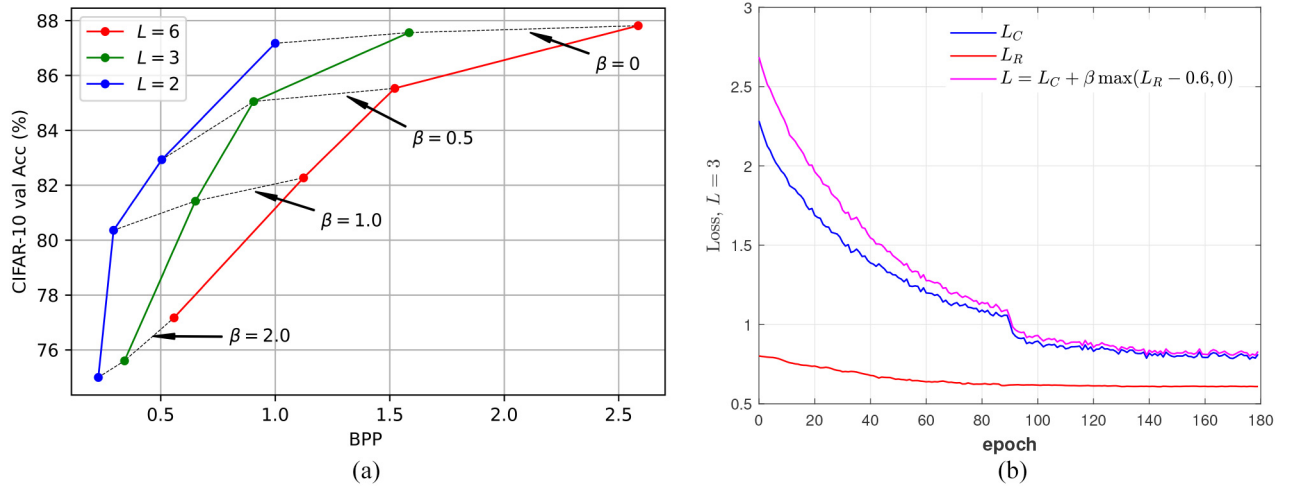


Fig. 13. (a) CIFAR-10 results for end-to-end compression and classification for different  $\beta$  values. Black dashed lines connect the points for AE models with the same  $\beta$  value. (b) CIFAR-10 average validation losses obtained with (33) at  $\beta = 2$  and  $r_t = 0.6$  for AE- $V_1$ ,  $L = 3$  model. Note the smooth convergence of the rate  $L_{CL}$  around 0.6 at higher number of epochs.

## VII. FURTHER THEORETICAL EXPLANATIONS

### A. Relationship to the Information Bottleneck

In this section, we theoretically derive the relationship between the proposed VAE framework for classification and the well-known IB principle [29], [30]. Consider the random variables  $\mathbf{Y}$ ,  $\mathbf{x}$ , and  $\mathbf{z}$  related according to a Markov chain as follows:

$$\mathbf{Y} \rightarrow \mathbf{x} \xrightarrow[\phi]{E} \mathbf{z}. \quad (34)$$

In the context of image classification with auto-encoders  $\mathbf{Y}$ ,  $\mathbf{x}$  and  $\mathbf{z}$  can be viewed as variables corresponding to image label, image, and latent encoding of the image, respectively.  $\mathbf{z}$  is mapped from  $\mathbf{x}$  with an Encoder ( $E$ ) parameterized by  $\phi$ . Hence, we can denote  $\mathbf{z}$  as  $\mathbf{z}(\phi)$ , which is often written as  $\mathbf{z}$  for simplicity.

We can write the objective function of the IB principle [29], [30] as the following:

$$\max_{\phi} \text{IB}_{\mathbf{x}, \mathbf{z}, \mathbf{Y}}(\phi) = I(\mathbf{z}(\phi); \mathbf{Y}) - \beta I(\mathbf{x}; \mathbf{z}(\phi)), \quad \beta \geq 0. \quad (35)$$

$I(\mathbf{x}; \mathbf{z})$  is the mutual information (MI) between the random variables  $\mathbf{x}$  and  $\mathbf{z}$ . We rewrite this as a minimization objective to match the above VAE narrative

$$\min_{\phi} -I(\mathbf{z}(\phi); \mathbf{Y}) + \beta I(\mathbf{x}; \mathbf{z}(\phi)), \quad \beta \geq 0. \quad (36)$$

Following the definition of MI we write:

$$I(\mathbf{z}; \mathbf{Y}) = \int p(y, \mathbf{z}) \log \frac{p(y|\mathbf{z})}{p(y)} dy d\mathbf{z}. \quad (37)$$

In order to track the distribution  $p(y|\mathbf{z})$  we employ a classifier (CL) parameterized by  $\theta$  and estimate the conditional label distribution  $\eta_{\theta}(y|\mathbf{z})$  as in (14). Since  $\text{KL}(p(\mathbf{Y}|\mathbf{z})|\eta_{\theta}(\mathbf{Y}|\mathbf{z})) \geq 0$  we can write the inequality

$$\int p(y|\mathbf{z}) \log p(y|\mathbf{z}) dy \geq \int p(y|\mathbf{z}) \log \eta_{\theta}(y|\mathbf{z}) dy. \quad (38)$$

We utilize this inequality to bound for  $I(\mathbf{z}; \mathbf{Y})$  in (37)

$$\begin{aligned} I(\mathbf{z}; \mathbf{Y}) &= \int p(y, \mathbf{z}) \log p(y|\mathbf{z}) dy d\mathbf{z} - \int p(y) \log p(y) dy \\ &\geq \int p(y, \mathbf{z}) \log \eta_{\theta}(y|\mathbf{z}) dy d\mathbf{z} + H(\mathbf{Y}). \end{aligned} \quad (39)$$

Leveraging the Markov assumption in (34),  $p(y, \mathbf{z}, \mathbf{x}) = p(\mathbf{x}, y)q_{\phi}(\mathbf{z}|\mathbf{x})$  with the encoder distribution  $q_{\phi}(\mathbf{z}|\mathbf{x})$ , we can express the bound in (39) for  $I(\mathbf{z}; \mathbf{Y})$  as the following:

$$\begin{aligned} I(\mathbf{z}; \mathbf{Y}) &\geq \int p(y, \mathbf{z}, \mathbf{x}) \log \eta_{\theta}(y|\mathbf{z}) dx dy d\mathbf{z} + H(\mathbf{Y}) \\ &\geq \int p(\mathbf{x}, y)q_{\phi}(\mathbf{z}|\mathbf{x}) \log \eta_{\theta}(y|\mathbf{z}) dx dy d\mathbf{z} + H(\mathbf{Y}). \end{aligned} \quad (40)$$

For the second term of the IB objective given in (36), following the definition of MI, we write:

$$\begin{aligned} I(\mathbf{x}; \mathbf{z}) &= \int q_{\phi}(\mathbf{x}, \mathbf{z}) \log q_{\phi}(\mathbf{z}|\mathbf{x}) d\mathbf{x} d\mathbf{z} \\ &\quad - \int q_{\phi}(\mathbf{z}) \log q_{\phi}(\mathbf{z}) d\mathbf{z}. \end{aligned} \quad (41)$$

With the definition of KL divergence, for an arbitrary  $r(\mathbf{z})$ , we write the following inequality similar to (38):

$$\int q_{\phi}(\mathbf{z}) \log q_{\phi}(\mathbf{z}) d\mathbf{z} \geq \int q_{\phi}(\mathbf{z}) \log r(\mathbf{z}) d\mathbf{z}. \quad (42)$$

We combine (38) and (41) to obtain an upper bound for  $I(\mathbf{x}; \mathbf{z})$

$$\begin{aligned} I(\mathbf{x}; \mathbf{z}) &\leq \int q_{\phi}(\mathbf{x}, \mathbf{z}) \log q_{\phi}(\mathbf{z}|\mathbf{x}) d\mathbf{x} d\mathbf{z} \\ &\quad - \int q_{\phi}(\mathbf{z}) \log r(\mathbf{z}) d\mathbf{z} \\ &= \int q_{\phi}(\mathbf{x}, \mathbf{z}) \log \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{r(\mathbf{z})} d\mathbf{x} d\mathbf{z} \\ &= \int p(\mathbf{x})q_{\phi}(\mathbf{z}|\mathbf{x}) \log \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{r(\mathbf{z})} d\mathbf{x} d\mathbf{z} \end{aligned} \quad (43)$$

$$= \int p(\mathbf{x}, y)q_{\phi}(\mathbf{z}|\mathbf{x}) \log \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{r(\mathbf{z})} dx dy d\mathbf{z}. \quad (44)$$

Equation (44) follows from (43) since  $p(\mathbf{x}) = \int p(\mathbf{x}, y) dy$ . Similar to [56], we approximate  $p(\mathbf{x}, y)$  with an empirical distribution based on data samples  $(\mathbf{x}_n, y_n) \in \mathcal{S}$  s.t.

$$p(\mathbf{x}, y) = \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \delta_{\mathbf{x}_n}(\mathbf{x}) \delta_{y_n}(y). \quad (45)$$

Hence, the approximated IB loss for minimization can be written as the following:

$$\begin{aligned} \text{IB}_{\mathbf{x}, \mathbf{z}, \mathbf{Y}}(\boldsymbol{\phi}, \boldsymbol{\theta}) &\leq - \int p(\mathbf{x}, y) q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \eta_{\boldsymbol{\theta}}(y|\mathbf{z}) d\mathbf{x} dy d\mathbf{z} - H(\mathbf{Y}) \\ &\quad + \beta \int p(\mathbf{x}, y) q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{r(\mathbf{z})} d\mathbf{x} dy d\mathbf{z} \end{aligned} \quad (46)$$

$$\begin{aligned} &\approx \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \left\{ - \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n) \log \eta_{\boldsymbol{\theta}}(y_n|\mathbf{z}) d\mathbf{z} \right. \\ &\quad \left. + \beta \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n) \log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n)}{r(\mathbf{z})} d\mathbf{z} \right\} - H(\mathbf{Y}) \end{aligned} \quad (47)$$

$$\begin{aligned} &= \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \left\{ - \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n) \log \eta_{\boldsymbol{\theta}}(y_n|\mathbf{z}) d\mathbf{z} \right. \\ &\quad \left. + \beta \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n) \log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n)}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_n)} d\mathbf{z} \right\} - H(\mathbf{Y}). \end{aligned} \quad (48)$$

From (47) to (48), we replace the arbitrary  $r(\mathbf{z})$  with  $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_n)$ .

Using the definition of KL divergence, we rewrite (48) as the following which is upper bounded by the  $\beta$ -VAE loss  $\mathcal{L}_{\boldsymbol{\theta}, \boldsymbol{\phi}}$  for classification:

$$\begin{aligned} \text{IB}_{\mathbf{x}, \mathbf{z}, \mathbf{Y}}(\boldsymbol{\phi}, \boldsymbol{\theta}) &\leq \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \left\{ - \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n) \log \eta_{\boldsymbol{\theta}}(y_n|\mathbf{z}) d\mathbf{z} \right. \\ &\quad \left. + \beta \text{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n), p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_n)] \right\} - H(\mathbf{Y}) \\ &\leq \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \{ \mathcal{L}_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{x}_n, y_n) \} - H(\mathbf{Y}). \end{aligned} \quad (49)$$

A similar result has been shown in [56] for image reconstruction.

### B. Alternative Problem Formulation

Assuming the same setting as we discussed above, we provide an alternative derivation the loss function for joint image classification and compression as follows.

We start with the classification CE loss between the true and estimated label distributions. From (18) and (19), we can write the CE loss as

$$\begin{aligned} \text{CE}_{y|\mathbf{x}}(\rho, \hat{\rho}_{\boldsymbol{\theta}}) &= - \sum_{c \in \mathcal{Y}} \rho(y = c|\mathbf{x}) \log \hat{\rho}_{\boldsymbol{\theta}}(y = c|\mathbf{x}) \\ &= - \log \hat{\rho}_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{x}). \end{aligned} \quad (50)$$

For the setting with an Encoder (**E**) with the density  $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ , we have

$$\mathbf{x} \xrightarrow[\boldsymbol{\phi}]{\text{E}} \mathbf{z} \xrightarrow[\boldsymbol{\theta}]{\text{CL}} y. \quad (51)$$

Considering different instances of  $\mathbf{z}$  for a given sample  $\mathbf{x}$ , (14) can be rewritten to formulate the following problem to maximize  $\hat{\rho}_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{x})$  in order to minimize classification CE:

$$\max_{\boldsymbol{\theta}, \boldsymbol{\phi}} E_{q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}} [\eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{z})]. \quad (52)$$

Since the density  $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$  is not available at the Decoder (**D**), we employ a PE  $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$  to closely estimate  $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ . This introduces a constraint to the problem in (52) with  $\epsilon \geq 0$  [26], [39]

$$\begin{aligned} &\max_{\boldsymbol{\theta}, \boldsymbol{\phi}} E_{q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}} [\eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{z})] \\ &\text{s.t. } 0 \leq \text{KL}(q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}}) \leq \epsilon. \end{aligned} \quad (53)$$

This can be rewritten as a minimization problem by introducing a monotonic  $-\log()$  function

$$\begin{aligned} &\min_{\boldsymbol{\theta}, \boldsymbol{\phi}} -\log E_{q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}} [\eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{z})] \\ &\text{s.t. } \text{KL}(q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}}) \leq \epsilon. \end{aligned} \quad (54)$$

According to Jensen's inequality, we can write

$$-\log E_{q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}} [\eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{z})] \leq E_{q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}} [-\log(\eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{z}))]. \quad (55)$$

Instead of directly minimizing the optimization problem in (54), we can minimize its upper bound as follows:

$$\begin{aligned} &\min_{\boldsymbol{\theta}, \boldsymbol{\phi}} E_{q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}} [-\log(\eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{z}))] \\ &\text{s.t. } \text{KL}(q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}}) \leq \epsilon. \end{aligned} \quad (56)$$

By rewriting the problem in (56) as a Lagrangian under KKT [57] conditions for the Lagrangian multiplier  $\beta \geq 0$ , we write

$$\begin{aligned} \mathcal{F}_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{x}, y_{\text{gt}}) &= E_{q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}} [-\log(\eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{z}))] \\ &\quad + \beta [\text{KL}(q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}}) - \epsilon]. \end{aligned} \quad (57)$$

With this we can arrive at the  $\beta$ -VAE loss function for classification

$$\begin{aligned} \mathcal{F}_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{x}, y_{\text{gt}}) &\leq \mathcal{L}_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{x}, y_{\text{gt}}) \\ &= E_{q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}} [-\log(\eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\mathbf{z}))] + \beta \text{KL}(q_{\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}}). \end{aligned} \quad (58)$$

## VIII. CONCLUSION

In this work, we propose a VAE-based end-to-end framework for joint compression and classification that autonomously learns on the latent features to efficiently compress and classify images in a networked AI edge/cloud environment. Starting from the classification CE loss, we present the theoretical foundation of the solution from an information theory perspective to define a rate-classification loss similar to rate-distortion in image reconstruction. Test results of our VAE learning networks on CIFAR-10, CIFAR-100, and ImageNet-1k data sets demonstrate significant improvement of image classification accuracy at the same bit rate. In particular, the proposed VAE-based joint compression and classification framework demonstrates over 10% and 4% classification accuracy improvement for CIFAR-10 and ImageNet-1k data sets, respectively, at data rate of 0.8 bpp in comparison with

the popular JPEG standard codec. We also achieve over  $\times 3$  bandwidth reduction for a given classification accuracy on the two data sets over JPEG. Extending the optimization of rate-classification performance of the proposed framework to address computation complexity is an interesting future direction to explore.

#### ACKNOWLEDGMENT

Our code is publicly available at <https://github.com/chamain/VAE-classifier>.

#### REFERENCES

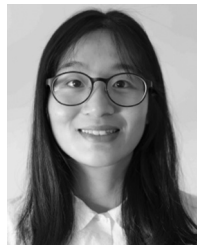
- [1] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)," in *Proc. Internet Technol. Appl. (ITA)*, 2015, pp. 219–224.
- [2] Q. Rao and J. Ftrunij, "Deep learning for self-driving cars: Chances and challenges," in *Proc. 1st Int. Workshop Softw. Eng. AI Auton. Syst.*, 2018, pp. 35–38.
- [3] G. Sreenu and M. A. S. Durai, "Intelligent video surveillance: A review through deep learning techniques for crowd analysis," *J. Big Data*, vol. 6, no. 1, p. 48, 2019.
- [4] Z. Yang and L. S. C. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," *Image Vis. Comput.*, vol. 69, pp. 143–154, Jan. 2018.
- [5] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. E. U. Haq, "Machine learning techniques for 5G and beyond," *IEEE Access*, vol. 9, pp. 23472–23488, 2021.
- [6] B. Ji *et al.*, "A survey of computational intelligence for 6G: Key technologies, applications and trends," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7145–7154, Oct. 2021.
- [7] C. De Alwis *et al.*, "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 836–886, 2021.
- [8] "Cisco visual networking index: Forecast and trends, 2017–2022," Cisco, San Jose, CA, USA, Rep. 1, vol. 1, 2018.
- [9] W. Yu *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2017.
- [10] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4921–4934, Jun. 2019.
- [11] A. A. Diro, N. Chilamkurti, and Y. Nam, "Analysis of lightweight encryption scheme for fog-to-things communication," *IEEE Access*, vol. 6, pp. 26820–26830, 2018.
- [12] C. Gong, F. Lin, X. Gong, and Y. Lu, "Intelligent cooperative edge computing in Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9372–9382, Oct. 2020.
- [13] H. Wang, C. Tao, J. Qi, H. Li, and Y. Tang, "Semi-supervised variational generative adversarial networks for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2019, pp. 9792–9794.
- [14] Y. Luo and H. Pfister, "Adversarial defense of image classification using a variational auto-encoder," 2018, *arXiv:1812.02891*.
- [15] X. Chen, Y. Sun, M. Zhang, and D. Peng, "Evolving deep convolutional variational autoencoders for image classification," *IEEE Trans. Evol. Comput.*, vol. 25, no. 5, pp. 815–829, Oct. 2021.
- [16] L. D. Chamain, S.-C. S. Cheung, and Z. Ding, "Quannet: Joint image compression and classification over channels with limited bandwidth," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2019, pp. 338–343.
- [17] L. D. Chamain and Z. Ding, "Faster and accurate classification for JPEG2000 compressed images in networked applications," 2019, *arXiv:1909.05638*.
- [18] L. D. Chamain and Z. Ding, "Improving deep learning classification of JPEG2000 images over bandlimited networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2020, pp. 4062–4066.
- [19] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *Proc. 8th Int. Conf. Qual. Multimedia Exp. (QoMEX)*, 2016, pp. 1–6.
- [20] L. D. Chamain, F. Racapé, J. Bégaïnt, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression for machines, a study," in *Proc. Data Compression Conf. (DCC)*, 2021, pp. 163–172.
- [21] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, 2021.
- [22] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/6316136/>
- [23] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, Feb. 1992.
- [24] E. Agustsson *et al.*, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Advances in Neural Information Processing Systems*, vol. 30. Red Hook, NY, USA: Curran, 2017.
- [25] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3214–3223. [Online]. Available: <http://dx.doi.org/10.1109/CVPR.2018.00339>
- [26] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.
- [27] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4394–4402.
- [28] G. Toderici *et al.*, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5306–5314.
- [29] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," 2000, *arXiv:physics/0004057*.
- [30] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2015, pp. 1–5.
- [31] *Information Technology—Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines*, Standard CCITT Rec. T.81, 1992.
- [32] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, vol. 642. New York, NY, USA: Springer, 2012.
- [33] A. Van Den Oord *et al.*, "WaveNet: A generative model for raw audio," in *Proc. SSW*, vol. 125, 2016, p. 2.
- [34] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," 2014, *arXiv:1410.8516*.
- [35] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2018, pp. 10215–10224.
- [36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*.
- [37] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*.
- [38] A. Habibiyan, T. V. Rozendaal, J. M. Tomczak, and T. S. Cohen, "Video compression with rate-distortion autoencoders," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7033–7042.
- [39] I. Higgins *et al.*, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–13.
- [40] S. Singh, S. Abu-El-Haija, N. Johnston, J. Ballé, A. Shrivastava, and G. Toderici, "End-to-end learning of compressible features," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2020, pp. 3349–3353.
- [41] X. Luo, H. Talebi, F. Yang, M. Elad, and P. Milanfar, "The rate-distortion-accuracy tradeoff: JPEG case study," 2020, *arXiv:2008.00605*.
- [42] M. Weber, C. Renggli, H. Grabner, and C. Zhang, "Observer dependent lossy image compression," 2019, *arXiv:1910.03472*.
- [43] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, "Faster neural networks straight from JPEG," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2018, pp. 3937–3948.
- [44] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero- and few-shot learning via aligned variational autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8247–8255.
- [45] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.
- [46] V. Sze and D. Marpe, "Entropy coding in HEVC," in *High Efficiency Video Coding (HEVC)*. Cham, Switzerland: Springer, 2014, pp. 209–274.

- [47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [48] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool, "Practical full resolution learned lossless image compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10629–10638.
- [49] A. Krizhevsky, V. Nair, and G. Hinton, 2014, "The CIFAR-10 Dataset," CIFAR-10. [Online]. Available: <http://www.cs.toronto.edu/kriz/cifar.html>
- [50] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.
- [51] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [52] Y. Deng, "Deep learning on mobile devices: A review," in *Proc. Mobile Multimedia Image Process. Security Appl.*, vol. 10993, 2019, Art. no. 109930A.
- [53] N. D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, and F. Kawsar, "Squeezing deep learning into mobile and embedded devices," *IEEE Pervasive Comput.*, vol. 16, no. 3, pp. 82–88, Jun. 2017.
- [54] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [55] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–16.
- [56] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.
- [57] H. W. Kuhn, "Nonlinear programming: A historical view," in *Traces and Emergence of Nonlinear Programming*. Basel, Switzerland: Springer, 2014, pp. 393–414.



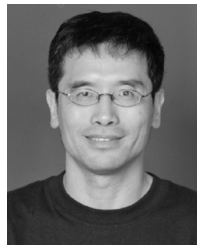
**Lahiru D. Chamain** (Student Member, IEEE) received the B.Sc. degree in electronics and telecommunication engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 2016, and the M.Sc. degree in electrical and computer engineering (ECE) from University of California at Davis, Davis, CA, USA, in 2020, where he is currently pursuing the Ph.D. degree in ECE.

His research interests align with image/video compression for deep learning, Internet of Things, and detection and estimation theory.



**Siyu Qi** received the B.Sc. degree in instrument science and engineering from Southeast University, Nanjing, China, in 2016. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA, USA.

Her research interests lie in the fields of image compression, deep learning, and time series forecasting.



**Zhi Ding** (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Cornell University, Ithaca, NY, USA, in 1990.

He is with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA, USA, where he currently a Distinguished Professor. From 1990 to 2000, he was a Faculty Member of Auburn University, Auburn, AL, USA, and later, University of Iowa, Iowa City, IA, USA. He has held visiting positions in Australian National University, Canberra, ACT,

Australia; Hong Kong University of Science and Technology, Hong Kong; NASA Lewis Research Center, Cleveland, OH, USA; and USAF Wright Laboratory. He has active collaboration with researchers from various universities in Australia, Canada, China, Finland, Hong Kong, Japan, Korea, Singapore, Taiwan, and USA. He has coauthored of the text: *Modern Digital and Analog Communication Systems*, (5th edition, Oxford University Press, 2019).

Prof. Ding received the IEEE Communication Society's WTC Award in 2012, and the IEEE Communication Society's Education Award in 2020. He was an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1994 to 1997 and from 2001 to 2004; and an Associate Editor of IEEE SIGNAL PROCESSING LETTERS from 2002 to 2005. He was the General Chair of the 2016 IEEE International Conference on Acoustics, Speech, and Signal Processing and the Technical Program Chair of the 2006 IEEE Globecom. He served on as IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS Steering Committee Member from 2007 to 2009, and its Chair from 2009 to 2010. He was a member of technical committee on Statistical Signal and Array Processing, and technical committee on Signal Processing for Communications from 1994 to 2003. He has been an active member of IEEE, serving on technical programs of several workshops and conferences. He was also an IEEE Distinguished Lecturer (Circuits and Systems Society from 2004 to 2006, and Communications Society from 2008 to 2009).