An End-to-End Learning Architecture for Efficient Image Encoding and Deep Learning

Lahiru D. Chamain, Siyu Qi, and Zhi Ding
Department of Electrical and Computer Engineering
University of California, Davis, CA, 95616
Email: {hdchamain, syqi, zding}@ucdavis.edu.

***Abstract*—Learning-based image/video codecs typically utilize the well known auto-encoder structure where the encoder transforms input data to a low-dimensional latent representation. Efficient latent encoding can reduce bandwidth needs during compression for transmission and storage. In this paper, we examine the effect of assigning high level coarse grouping labels to each latent vector. Designing coding profiles for each latent group can achieve high compression encoding. We show that such grouping can be learned via end-to-end optimization of the codec and the deep learning (DL) model to optimize rate-accuracy for a given data set. For cloud-based inference, source encoder can select a coding profile based on its learned grouping and encode the data features accordingly. Our test results on image classification show that significant performance improvement can be achieved with learned grouping over its non-grouping counterpart.**

***Index Terms*—Grouping, end-to-end encoding, classification.**

I. Introduction

Deep learning applications on images and video data generated by distributed low end devices are continuously expanding at a staggering pace. In such networked artificial intelligence (AI) scenarios, low end devices such as roadside cameras, vehicle sensors, and IoT devices are in charge of data capturing before transporting them to cloud/edge servers with high memory and computational capacity required for executing machine tasks. Under limited (wireless) network bandwidth, image/video data must be compressively encoded for transport channels without sacrificing machine learning (ML) task accuracy or visualization quality at the remote end [1]–[3].

Another important consideration in deep learning is the reliability of training data. Inference performance of supervised learning tasks such as image classification, object recognition, and segmentation, depends critically on the accuracy of labeled data available for training [4]. Training samples can be mislabelled due to human errors and occasional corruptions during transmission and storage. Hence, an equally important problem in distributed ML is to learning algorithms, robust to training mislabels [5].

Recently developed image/video codecs based on deep learning often feature an auto-encoder structure [6]–[8]. The encoder maps the high dimensional ($\mathbb{R}^D$) input manifold/space of high complexity to a low dimensional ($\mathbb{R}^d$) latent representation. Key features of the latent representation are acquired from end-to-end optimization including the codec and media processing model. This work addresses both aforementioned problems and optimizes the latent representation for efficient encoding and effective ML.

We organize the paper as follows. In Section II, we review a recent proposal of Maximal Coding Rate Reduction (MCR^2) principle of latent optimization in supervised learning and its robustness to mislabeled training data. We build a direct connection of MCR^2 with efficient latent encoding for transmission in the distributed deep learning paradigm. We propose an end-to-end optimization of networked image classification system by leveraging the concept of data grouping in MCR^2 in Sec. III. Finally, we present test results on CIFAR-100 and ImageNet-1K (size 32) data sets in Sec. IV.

II. Background and Related works

Let $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_1}, \cdots \boldsymbol{x_m}] \in \mathbb{R}^{D\times m}$ be m i.i.d. samples of dimension D. An encoder $f(\cdot, \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$ maps each sample $\boldsymbol{x}$ to a d-dimensional ($d < D$) learned representation $\boldsymbol{z}$ such that $\boldsymbol{z} = f(\boldsymbol{x}, \boldsymbol{\theta})$. We write the set of latent vectors mapped from set $\boldsymbol{X}$ as $\boldsymbol{Z} = [\boldsymbol{z_1}, \boldsymbol{z_2}, \cdots, \boldsymbol{z_m}] \in \mathbb{R}^{d\times m}$.

$$\boldsymbol{x} \xrightarrow[f(\cdot,\boldsymbol{\theta})]{\text{Encoder}} \boldsymbol{z}(\boldsymbol{\theta}) \xrightarrow{\text{Channel}} \hat{\boldsymbol{z}}(\boldsymbol{\theta}) \xrightarrow[g(.,\boldsymbol{\phi})]{\text{Classifier}} y \tag{1}$$

Source device uses entropy coding to encode $\boldsymbol{z}$ for transmission. The receiver decodes the code words to obtain reconstructed $\hat{\boldsymbol{z}}$. An ML task model $g(\cdot, \boldsymbol{\phi})$ with parameters $\boldsymbol{\phi}$ generates the output label $y = g(\hat{\boldsymbol{z}}, \boldsymbol{\phi})$ based on $\hat{\boldsymbol{z}}$.

A. Non-asymptotic rate-distortion with multiple groups

The authors of [9] provided a tight upper bound on the number of bits required to encode $\boldsymbol{X}$ in a subspace. For a Gaussian channel with distortion ϵ^2, defined as the ℓ_2-norm of reconstruction error, the mean code length per sample (for large m) is [9]:

$$r(\boldsymbol{Z}|\epsilon) = \frac{1}{2}\log_2 \det\left[\boldsymbol{I} + \frac{d}{m\epsilon^2}\boldsymbol{Z}\boldsymbol{Z}^\top\right] \text{ bits.} \tag{2}$$

This result requires $\boldsymbol{Z}$ to be within the same d-dim subspace. Partition $\boldsymbol{Z}$ into k disjoint subsets (groups) based on features: $\boldsymbol{Z} = \boldsymbol{Z}_1 \cup \cdots \cup \boldsymbol{Z}_k$. The grouping of $\boldsymbol{Z}$ can be denoted by a membership set $\boldsymbol{\Pi}$ of binary diagonal matrices $\{\boldsymbol{\Pi}_1, \cdots, \boldsymbol{\Pi}_k\} \in \mathbb{R}^{m\times m}$ with $\sum_{i=1}^k \boldsymbol{\Pi}_i = \boldsymbol{I}_m$. Each group

has size $\mathrm{tr}(\boldsymbol{\Pi}_j)$. For this case, [9] similarly provided a tight upper bound of average rate (bits per sample) when $m \gg d$:

$$r^c(\boldsymbol{Z}|\epsilon, \boldsymbol{\Pi}) = \sum_{j=1}^{k} \frac{\mathrm{tr}(\boldsymbol{\Pi}_j)}{2m} \log_2 \det \left[\boldsymbol{I} + \frac{d \cdot \boldsymbol{Z}\boldsymbol{\Pi}_j \boldsymbol{Z}^\top}{\mathrm{tr}(\boldsymbol{\Pi}_j)\epsilon^2} \right] \quad (3)$$

B. Maximal coding for rate reduction (MCR^2)

To find good a representation of $\boldsymbol{Z}(\boldsymbol{\theta}) = f(\boldsymbol{X}, \boldsymbol{\theta})$, the MCR^2 principle of [10] maximizes the loss function of

$$\Delta r(\boldsymbol{Z}(\boldsymbol{\theta})|\epsilon, \boldsymbol{\Pi}) = r(\boldsymbol{Z}(\boldsymbol{\theta})|\epsilon) - r^c(\boldsymbol{Z}(\boldsymbol{\theta})|\epsilon, \boldsymbol{\Pi}). \quad (4)$$

As shown in [10], the first term of (4) measures the code length for all features in $\boldsymbol{Z}$ whereas the second term is the sum code length of features in each of the k groups, Treating $\boldsymbol{z}$ as the output of the final feature layer in MCR^2, [10] used a simple subspace classifier after applying true class labels to generate the membership set $\boldsymbol{\Pi}$. This MCR^2 classifier is robust against mislabeled images during training.

C. MCR^2 and Latent Encoding

Cloud-based deep learning (DL) applications involve low-end devices to capture images and videos for encoding and transmitting to powerful computing nodes to carry out learning. To reduce power and bandwidth consumption, it is vital to efficiently encode the latent $\boldsymbol{Z}$ by minimizing $r^c(\boldsymbol{Z}(\boldsymbol{\theta})|\epsilon, \boldsymbol{\Pi})$ for transmission.

Instead of only minimizing $\Delta r(\boldsymbol{Z}(\boldsymbol{\theta})|\epsilon, \boldsymbol{\Pi})$ as in MCR^2, the need to improve code efficiency by constraining $r^c(\boldsymbol{Z}(\boldsymbol{\theta})|\epsilon, \boldsymbol{\Pi})$ is practically significant. In fact, maximizing the MCR^2 objective in Eq. (4) does not necessarily guarantee to reduce $r^c(\boldsymbol{Z}(\boldsymbol{\theta})|\epsilon, \boldsymbol{\Pi})$. In this work, we focus on the dual objective of achieving robustness against mislabeling and reducing the rate $r^c(\boldsymbol{Z}(\boldsymbol{\theta})|\epsilon, \boldsymbol{\Pi})$ of efficient latent encoding by leveraging the grouping information.

D. Latent Compression in View of Grouping

Several previous works utilized grouping for better rate-distortion performance in image/video compression [11]–[13]. The authors of [11] proposed dividing videos into clusters, each with distinct encoding profile in video compression for transmission. For image compression, the authors of [13] proposed to encode highly correlated images together to improve overall compression ratio. Similarly, we explore the benefits of the grouping to extract compressive features for a given DL task under bandwidth constraint.

Optimizing image/video compression codecs for a given task is an active area of research. Some recent works [1]–[3] showed promises by jointly optimizing the codec and DL model including the entropy coding parameters. Since the learned representation $\boldsymbol{Z}$ is not unique, the encoder $\boldsymbol{\theta}$ and task model $\boldsymbol{\phi}$ can be end-to-end optimized for a learning objective such as classification or segmentation.

Grouping information helps generate membership set $\boldsymbol{\Pi}$. Candidates for grouping information include ground truth labels, coarse labels and tags (metadata), depending often on the learning task. For instance, in hierarchical classification, coarse labels have shown benefits for improving classification accuracy [14], [15]. Similarly, Metadata are less prone to error compared to ground truth labels since accurate manual labelling is not needed.

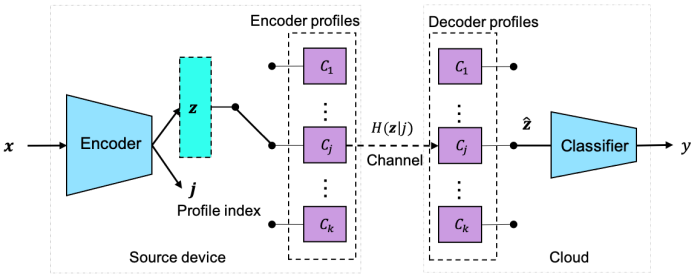


Fig. 1. Proposed framework. During inference, for input image $\boldsymbol{x}$, the encoder generates a profile index j and the feature vector $\boldsymbol{z}$. Then, the encoder profile of C_j encodes $\boldsymbol{z}$ to a bit stream with entropy $\mathrm{H}(\boldsymbol{z}|j)$.

In practice, initial grouping information may not be available at acquisition or may not be a good candidate for a given task. To be broadly accommodating, we propose that the grouping be learned through unsupervised learning for end-to-end optimization to achieve better rate-accuracy trade-off. In real time applications, source device can then select a pre-trained coding profile per group based on the rate-accuracy trade-off.

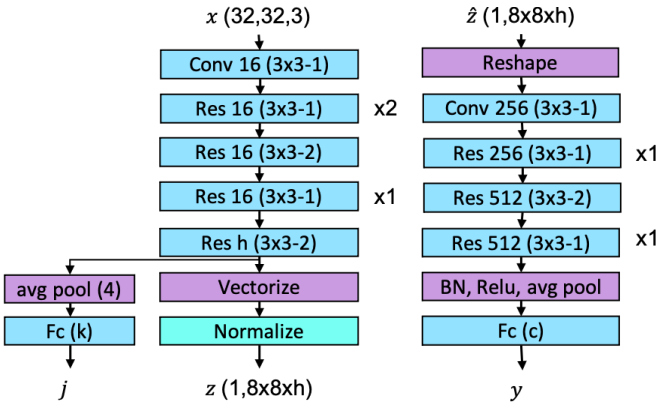
III. Proposed End-to-End Framework

Our proposed framework in Fig. 1 consists of an encoder, a set of coding profiles, and a task model (e.g. an image classifier). Encoder maps image $\boldsymbol{x}$ to a low-order latent vector $\boldsymbol{z}$ and generates a group label $j \in \{1, 2, \cdots, k\}$, referred to as the "profile index". This profile set has k different encoding-decoding profiles $\{C_j\}_1^k$, each of which is optimized to compress latents with profile index j. For image encoding, a profile may typically consist of a quantizer, an entropy coder, an entropy decoder and an optional de-quantizer. Classifier uses the decoded latent $\hat{\boldsymbol{z}}$ as input for classification into c classes with $c \geq k$.

During training of the end-to-end architecture, parameters of the encoder, coding profile, and the classifier are jointly optimized. The optimized encoder and decoder profiles are stored at the source and the cloud nodes, respectively. During inference phase, the source encoder determines the profile index j for the input image $\boldsymbol{x}$ and generates the feature vector $\boldsymbol{z}$. Next, the encoder uses profile C_j to encode $\boldsymbol{z}$ into bit stream for transmission to the classifier node on cloud/edge. The classifier decodes the received bit stream based on profile C_j to recover the feature vector $\hat{\boldsymbol{z}}$ for subsequent classification.

A. Rate under Quantization Noise

Quantization is the primary source of rate reduction in both commercial [16] and recent DL based image/video codes [6], [8]. Hence, we model the wireless channel with quantization noise $\boldsymbol{n} \in \mathbb{R}^d$ as follows. Let $\boldsymbol{z} \in \mathbb{R}^d$ be the recovered vector from the quantized vector $\hat{\boldsymbol{z}} \in \mathbb{R}^d$. Then it is clear that $\hat{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{n}$. Following previous works [2], [6], for quantization



(a) Encoder (b) Classifier

Fig. 2. Encoder and classifier architectures used for CIFAR-100 ($c = 100$) and ImageNet-1K ($c = 1000$). Number of filters of the last ResNet block of the encoder are $h = 16, 12$ for hidden sizes $d = 1024, 768$, respectively. "Conv 16 (3x3-2)" represents a 2D convolution block with 16 filters of size 3×3 and stride of 2. "Res 16 (3x3-2)" represents a basic ResNet block [17] with down-sampling factor 2.

step s, elements of $\boldsymbol{n}$ can be modeled as independent, zero mean, and uniformly distributed in $[-s/\sqrt{d}, s/\sqrt{d}]$. Hence, the distortion ϵ from quantization noise forms an upper bound of reconstruction error:

$$\mathbb{E}[||\boldsymbol{z} - \hat{\boldsymbol{z}}||_2] \leq \epsilon^2 = \frac{(2s)^2}{12} \tag{5}$$

Following [9], we can derive an upper bound for the rate at a given distortion ϵ using the "sphere packing" principle [18] in information theory. To begin, estimate covariance matrix

$$\hat{\boldsymbol{\Sigma}} = \mathbb{E}[\frac{1}{m}\sum_{i=1}^{m} \hat{\boldsymbol{z}}_i \hat{\boldsymbol{z}}_i^\top] = \frac{1}{m}\boldsymbol{Z}\boldsymbol{Z}^\top + \frac{\epsilon^2}{d}\boldsymbol{I}_d. \tag{6}$$

The volume spanned by the vectors $\hat{\boldsymbol{z}}$ is upper-bounded by the volume of vectors with Gaussian density of same covariance.

$$\mathrm{vol}(\hat{\boldsymbol{Z}}) \leq \sqrt{(2\pi e)^d \det \hat{\boldsymbol{\Sigma}}} \tag{7}$$

Similarly, the volume spanned by the uniform noise is

$$\mathrm{vol}(\boldsymbol{N}) = \left(\frac{2s}{\sqrt{d}}\right)^d = \sqrt{\det \frac{(2s)^2}{d}\boldsymbol{I}_d}. \tag{8}$$

The number of bits to represent each vector $\boldsymbol{z}$ satisfying the ℓ_2-error bound ϵ^2 can be found as the number of bits to represent the index of a sphere spanned by uniform noise, packed in the region spanned by $\hat{\boldsymbol{\Sigma}}$. Therefore, we express an upper bound for average rate of a sample $\boldsymbol{z}$ in bits at a distortion ϵ under uniform quantization noise as

$$r(\boldsymbol{Z}|\epsilon) = \frac{1}{2}\log_2 \det\left[\boldsymbol{I} + \frac{d}{m\epsilon^2}\boldsymbol{Z}\boldsymbol{Z}^\top\right] + \frac{d}{2}\log_2\left[\frac{2\pi e}{12}\right]. \tag{9}$$

Note that the rate depends only on singular values of $\boldsymbol{Z}$. Compared to the rate under Gaussian noise given in Eq. (2), the upper bound for the rate under quantization noise only has an additional linear term of d.

B. Learning to Group

Similar to Sec. II-A, the set $\boldsymbol{Z}$ can be partitioned to k subsets $\boldsymbol{Z}_1 \cdots \boldsymbol{Z}_k$ according to a membership set $\boldsymbol{\Pi}$ of diagonal matrices $\{\boldsymbol{\Pi}_1, \cdots, \boldsymbol{\Pi}_k\} \in \mathbb{R}^{m\times m}$. Each diagonal element of $\boldsymbol{\Pi}_j(i,i)$ denotes the probability of sample i belonging to group j. Note that each sample can only belong to one group [9]:

$$\boldsymbol{\Pi}_j(i,i) = \pi_{ij} \in [0,1], \quad \textstyle\sum_{j=1}^{k} \pi_{ij} = 1.$$

Group j contains $m_j = \mathrm{tr}(\boldsymbol{\Pi}_j)$ samples. Similar to Eq. (3), for $m \gg d$, an upper bound for average rate for the given grouping $\boldsymbol{\Pi}$ can be written as

$$r^c(\boldsymbol{Z}|\epsilon, \boldsymbol{\Pi}) = \sum_{j=1}^{k} \frac{\mathrm{tr}(\boldsymbol{\Pi}_j)}{2m}\log_2 \det\left[\boldsymbol{I} + \frac{d}{\mathrm{tr}(\boldsymbol{\Pi}_j)\epsilon^2}\boldsymbol{Z}\boldsymbol{\Pi}_j\boldsymbol{Z}^\top\right] + \frac{d}{2}\log_2\left[\frac{2\pi e}{12}\right]. \tag{10}$$

From (10), an optimal grouping $\boldsymbol{\Pi}^*$ can be learned to minimize the average rate-distortion for each sample $\boldsymbol{z}$. Thus, we formulate a rate minimization problem for given distortion ϵ:

$$\boldsymbol{\Pi}^* = \arg\min_{\boldsymbol{\Pi}} r^c(\boldsymbol{Z}|\epsilon, \boldsymbol{\Pi}) \tag{11}$$

We extend this rate minimization to an end-to-end optimization problem in conjunction with a learning task. Consider image classification using model $g(\cdot, \boldsymbol{\phi})$ that takes the reconstructed vector $\hat{\boldsymbol{z}}$ as the input and generates output label $y = g(\hat{\boldsymbol{z}}, \boldsymbol{\phi})$ as shown in Eq. (1). We propose to learn $\boldsymbol{\Pi}^*$ via end-to-end optimization of encoder f and learning model g. For this purpose, we minimize the following loss function.

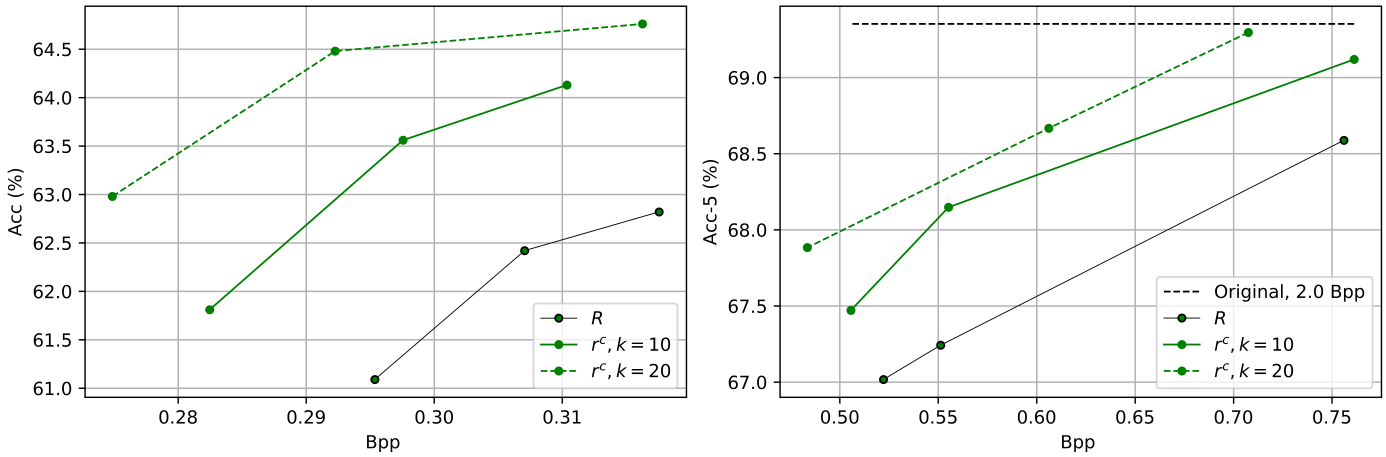
$$\boldsymbol{\Pi}^*, \boldsymbol{\theta}^*, \boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\Pi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \epsilon} \mathbb{E}[\mathcal{L}(y, y_{gt})] + \lambda \cdot r^c(\boldsymbol{Z}|\epsilon, \boldsymbol{\Pi}) \tag{12}$$

$\mathcal{L}(y, y_{gt})$ denotes the classification cross entropy loss between the inferred class label y and the ground truth label y_{gt}. Here, the first RHS term of Eq. (12) is the average classification loss. λ is the Lagrangian to manage the trade-off between the rate and task accuracy [2], [3], [19]. $\boldsymbol{\theta}^*$ and $\boldsymbol{\phi}^*$ denote the learned encoder and classifier parameters, respectively.

We also propose to make the distortion variable ϵ trainable. This relaxation enables encoder more degree of freedom to adjust rate and shifts the role of controlling rate of latent vectors to parameter λ. On the other hand, since ϵ is related to the quantization parameter s, this allows the network to optimize the quantization interval as well.

IV. Experiments and Results

In this section, we describe the experimental setup and present test results from CIFAR-10 [20], CIFAR-100 [21] and ImageNet-1k [22] data sets. CIFAR-10 consists of 10 classes with 5000 training and 1000 test RGB images of size 32×32 per class. ImageNet-1k (size 32) consists of 1000 classes, each containing up to 1300 training images and 50 validation images resized to 32×32. Similar to CIFAR-10, CIFAR-100 data set contains 50k training and 10k test RGB images of size 32×32 in 100 classes.



(a) CIFAR-100 (b) ImageNet-1k (size 32)

Fig. 3. Rate-accuracy performance for CIFAR-100 ($d = 768$) and ImageNet-1k ($d = 1024$) data sets under quantization noise.

A. DL Network Architectures

Fig. 2 describes the encoder and classifier architectures in use. To generate the profile index j, we used a simple fully-connected layer with k nodes and assigned the index of the largest node as j.

For experiments that include a bandwidth/rate constraint, we initialized the training of end-to-end encoder-classifier framework shown in Fig. 1 by using pre-trained models, trained without a rate constraint, i.e., pre-trained models that were optimized for classification only. In CIFAR-10 and CIFAR-100 training, we fine-tuned the model using a "stochastic gradient decent" optimizer with a initial learning rate of 0.05. We reduced the learning rate each time by $\times 0.1$ at 10, 20, and 30 epochs until termination at 40 epochs. Similarly for ImageNet-1k (size 32), we used the same optimizer starting from a learning rate of 0.05, reduced each time by $\times 0.1$ at 5 and 10 epochs, respectively, until termination at 15 epochs[1].

B. Quantization Noise Emulation

We added random uniform noise to the vector $\boldsymbol{z}$ to generate $\hat{\boldsymbol{z}}$ as the input to the classifier according to Sec. III-A. Following the work [2], we applied rounding function during the forward pass of the training and no quantization during the backward pass to approximate the loss function gradient. Consider an element z of the vector $\boldsymbol{z}$. We can write the differentiable quantization operation that maps z to quantized $\hat{z}$ in *Pytorch* as

$$\hat{z} = \text{torch.round}(\frac{z}{S}) \times S - z.\text{detach}() + z, \qquad (13)$$

where $S = 2s/\sqrt{d}$ is the stepsize in this implementation.

Fig. 3 provides the results under quantization noise for CIFAR-100 and ImageNet-1k (size 32) data sets respectively for 10 and 20 coding profiles. We record the rate in bits per pixel (Bpp). With 20 learned coding profiles CIFAR-100 shows over 3.5% classification accuracy improvements over no grouping at 0.295 Bpp. Similarly, ImageNet-1k shows over 1% top-5 accuracy improvements at 0.525 bpp with 20 coding profiles over no grouping. We further note that increasing the number of coding profiles improves rate-accuracy tradeoff for both data sets particularly at lower data rates.

C. Ablation Experiment: Distortion Learning

In this study, we test trainable distortion parameter ϵ. By making ϵ trainable, we allow the network to optimize quantization stepsize s. To observe the benefit of trainable distortion ϵ, we also generate rate-accuracy points by optimizing the proposed framework for fixed ϵ values, as baselines. From the results in Fig. 4, we see that trainable ϵ makes it easier to change rate without having to set ϵ manually for each rate. Further, trainable ϵ produces comparably good results to fixed ϵ case.

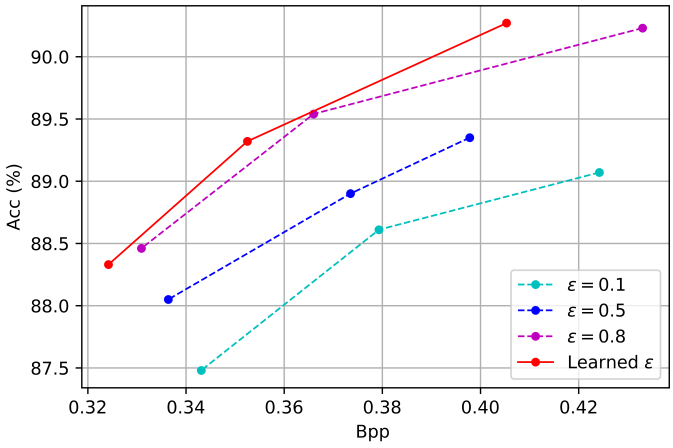


Fig. 4. Learned distortion: Ablation results using CIFAR-10 ($d = 1024$) data set without grouping (k=1).

[1]Source code is given at https://github.com/chamain/Learning-to-group.

D. Gaussian noise

The performance gains achieved with the proposed grouping and end-to-end optimization are consistent under Gaussian noise. In this experiment, we added random Gaussian noise to $\boldsymbol{z}$ to generate noisy $\hat{\boldsymbol{z}}$ as the input to classifier. To train the network, we used the same function of Eq. (12) and updated the rate term $r^c(\boldsymbol{Z}|\epsilon, \boldsymbol{\Pi})$ for Gaussian noise as in Eq. (3). Considering practical applications, we fixed ϵ at 0.5 during the training. For the encoder-classifier model, we used the same optimizer, learning rates and scheduling as described in Sec. IV-A.

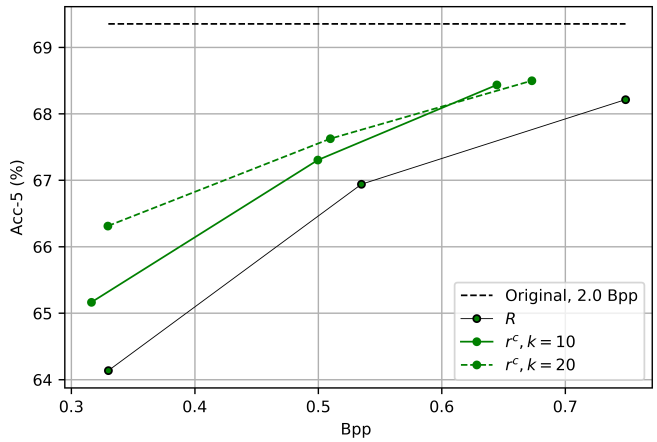


Fig. 5. Rate-accuracy performance for ImageNet-1k ($d = 1024$) data set under Gaussian noise.

For ImageNet-1k (size 32) data set, results from Fig. 5 confirms the rate-accuracy performance gains with the latent dimensions $d = 1024$ for coding profiles of 10 and 20, respectively. With 20 learned coding profiles, the proposed method achieves over 2% top-5 classification accuracy improvement at 0.330 Bpp. We further note that these results of rate-accuracy performance improvement are consistent with results from quantization noise tests.

V. Conclusions

In this paper, we proposed a grouping-based end-to-end compression and classification framework for distributed learning involving low cost sensing devices. Based on CIFAR and ImageNet data sets, we observed considerable rate-classification accuracy improvements with learned grouping compared to no grouping case under quantization and Gaussian noise cases. The achieved rate-accuracy gains with learned grouping increase with number of grouping profiles at the cost of higher encoder complexity. We further note that the proposed architecture is computationally simple and easily trainable. In future works, we plan to explore more effective methods to accurately generate coding profile index. Equally important is the work to define discriminative group labels that are also simultaneously compressive.

References

[1] L. D. Chamain, F. Racapé, J. Bégaint, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression for machines, a study," in *2021 Data Compression Conference (DCC)*. IEEE, 2021, pp. 163–172.

[2] S. Singh, S. Abu-El-Haija, N. Johnston, J. Ballé, A. Shrivastava, and G. Toderici, "End-to-end learning of compressible features," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3349–3353.

[3] R. Torfason, F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Towards image understanding from deep compression without decoding," *arXiv preprint arXiv:1803.06131*, 2018.

[4] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels." in *NIPS*, vol. 26, 2013, pp. 1196–1204.

[5] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5552–5560.

[6] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[7] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4394–4402.

[8] A. Habibian, T. v. Rozendaal, J. M. Tomczak, and T. S. Cohen, "Video compression with rate-distortion autoencoders," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7033–7042.

[9] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 9, pp. 1546–1562, 2007.

[10] Y. Yu, K. H. R. Chan, C. You, C. Song, and Y. Ma, "Learning diverse and discriminative representations via the principle of maximal coding rate reduction," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[11] H. Wei, Y. Yang, L. Li, A. B. Winston, and A. Ten-Ami, "Hybrid learning for adaptive video grouping and compression," Sep. 17 2019, uS Patent 10,419,773.

[12] K.-O. Cheng, N.-F. Law, and W.-C. Siu, "Clustering-based compression for population dna sequences," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 1, pp. 208–221, 2017.

[13] R. Kozhemiakin, S. Abramov, V. Lukin, B. Djurović, I. Djurović, and B. Vozel, "Lossy compression of landsat multispectral images," in *2016 5th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2016, pp. 104–107.

[14] X. Zhu and M. Bain, "B-cnn: branch convolutional neural network for hierarchical classification," *arXiv preprint arXiv:1709.09890*, 2017.

[15] J. Y. Chang and K. M. Lee, "Large margin learning of hierarchical semantic similarity for image classification," *Computer Vision and Image Understanding*, vol. 132, pp. 3–11, 2015.

[16] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conf. on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[19] L. D. Chamain and Z. Ding, "Improving deep learning classification of jpeg2000 images over bandlimited networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4062–4066.

[20] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[21] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.