

# An end-to-end joint learning scheme of image compression and quality enhancement with improved entropy minimization

Jooyoung Lee<sup>1,2</sup> | Seunghyun Cho<sup>3</sup> | Munchurl Kim<sup>1</sup> 

<sup>1</sup>School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea

<sup>2</sup>Media Research Division, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

<sup>3</sup>Space Technology Center, Agency for Defense Development, Daejeon, Republic of Korea

## Correspondence

Munchurl Kim, School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea.  
Email: [mkimee@kaist.ac.kr](mailto:mkimee@kaist.ac.kr)

## Funding information

Institute for Information & Communications Technology Planning & Evaluation (IITP), Grant/Award Number: 2017-0-00072

## Abstract

Recently, learned image compression methods based on entropy minimization have achieved superior results compared with conventional image codecs such as BPG and JPEG2000. However, they leverage single Gaussian models, which have a limited ability to approximate various irregular distributions of transformed latent representations, resulting in suboptimal coding efficiency. Furthermore, existing methods focus on constructing effective entropy models, rather than utilizing modern architectural techniques. In this paper, we propose a novel joint learning scheme called JointIQ-Net that incorporates image compression and quality enhancement technologies with improved entropy minimization based on a newly adopted Gaussian mixture model. We also exploit global context to estimate the distributions of latent representations precisely. The results of extensive experiments demonstrate that JointIQ-Net achieves remarkable performance improvements in terms of coding efficiency compared with existing learned image compression methods and conventional codecs. To the best of our knowledge, ours is the first learned image compression method that outperforms VVC intra-coding in terms of both PSNR and MS-SSIM.

## KEYWORDS

deep learning, entropy minimization, learned image compression

## 1 | INTRODUCTION

Recently, significant progress in the development of artificial neural networks (ANNs) has led to groundbreaking achievements in various fields of research. In the image and video compression domain, a number of learning-based studies [1–18] have been conducted. Specifically, several recent end-to-end optimized image compression approaches [6, 10] based on entropy minimization have

already achieved better compression performance than existing image compression codecs such as BPG [19] and JPEG2000 [20], despite the short history of the field. The basic approach to entropy minimization is to train analysis (encoder) and synthesis (decoder) transformer networks, allowing them to reduce the entropy of transformed latent representations while maintaining the quality of the reconstructed images to ensure that they remain as close as possible to the original images, where

entropy is calculated based on the distribution models of the transformed latent representations, which are called entropy models. Therefore, the distribution modeling of latent representations is the key element of entropy minimization and allows an entropy model to approximate the actual entropy of latent representations. Existing methods [2, 6, 10] adopt single Gaussian-based entropy models but have limited ability to approximate various irregular distributions of latent representations. To address this issue, we propose an improved Gaussian mixture model (GMM)-based entropy model and demonstrate that it significantly improves coding efficiency by estimating the distributions of latent representations more precisely.

Based on the assumed distribution models of latent representations, an image compression method estimates model parameter values such as  $\mu$  and  $\sigma$  in the case of a single Gaussian model for each representation element. During this process, additional information such as previously decoded neighbor representation values or certain bit-allocated side information can be used as contextual information for estimation. Contextual information can be considered as information provided to the model parameter estimator that helps predict the distributions of latent representations more precisely. Previous autoregressive approaches [6, 10] have used locally adjacent representations as additional context. However, they have not investigated how to utilize globally scattered representations. In this paper, we define a new type of global context (GC) utilizing information aggregated from the global area of latent representations and demonstrate that it can further improve coding efficiency by removing remaining spatial correlations across a wider area of representations. This GC exploitation is motivated by the concept of known wisdom [21, 22], which exploits the self-similarity in input images. Although the proposed GC exploitation yields only a marginal improvement in coding efficiency, it is meaningful because it is the first attempt to utilize global information directly to estimate the distributions of latent representations.

From an architectural perspective, we propose a simple but highly effective scheme that incorporates image compression and quality enhancement techniques. Recent approaches [2, 6, 10] in the learned image compression domain have focused on constructing effective entropy models, rather than utilizing modern architectural techniques. In the area of quality enhancement, research is continuously being conducted from an architectural perspective, yielding superior results compared with those of traditional methods, as described in Section 2. To leverage the advantages of both image compression and quality enhancement approaches, we connected image compression and quality enhancement

subnetworks in a cascade and jointly trained them to realize improved coding efficiency. Through various experiments, we determined that this unified scheme significantly improves coding efficiency. The key contributions of this study can be summarized as follows.

- We propose a novel end-to-end learning scheme called JointIQ-Net that incorporates both image compression and quality enhancement in a unified architecture.
- We propose an improved entropy-minimization method that uses a GMM for the distribution modeling of latent representations, whose parameters are accurately estimated by an improved estimator trained through the joint optimization of image compression and quality enhancement, yielding improved coding efficiency.
- To improve entropy minimization further, we utilize a GC when estimating GMM parameters, capturing a broader range of contextual information and helping reduce the spatial correlations between the current latent representation and other representations in a nonlocal range.
- To the best of our knowledge, JointIQ-Net is the *first* deep image compression scheme that outperforms VVC intra-coding (VTM 7.1 [23]) in terms of both PSNR and MS-SSIM and also yields significant improvements over existing learned image compression approaches.

## 2 | RELATED WORK

Recently, several learned image and video compression methods [1, 2, 4–12, 14–17] have been actively studied to exploit superior transformation capabilities based on the nonlinearity of neural networks. In contrast, traditional codecs [19, 20, 24] and approaches [25–29] still utilize linear transforms such as discrete cosine transform (DCT) or principal component analysis (PCA). In the research area of learned image compression, Toderici and others [16, 17] first introduced a novel image compression method using a small number of latent binary representations, which can improve image quality in a progressive manner. Johnston and others [4] enhanced the network operation method proposed by Toderici and others [17] to achieve better coding efficiency. Meanwhile, Ballé and others [1] and Theis and others [15] introduced novel image compression methods based on the entropy minimization of latent representations. Ballé and others [2] enhanced their entropy model by adopting a hierarchical prior model to estimate the distributions of latent representations in an input-adaptive manner, whereas former approaches [1, 15] learned the distribution parameters of latent representations directly. Therefore, those

parameters were fixed during the inference step. Minnen and others [10] and Lee and others [6] proposed autoregressive models that utilize adjacent latent representations as additional contextual information to estimate the distribution of the current representation. Both approaches enhance compression performance and yield better results than BPG [19], which is an image compression codec based on HEVC (ISO/IEC 23008-2, ITU-T H.265) intra-coding [30]. Several variations [11, 31] have been proposed to train autoregressive models that exploit cross-channel correlations in latent representations. He and others [3] utilized a checkerboard-type context to boost encoding and decoding speed while exploiting the excellent coding efficiency of the autoregressive model. In more recent studies [13, 18], Zhu and others [18] replaced all transform convolutions with Swin Transformer [32] blocks and Qian et al. [13] utilized a self-attention stack to replace hyper-encoder/decoder networks.

ANN-based image restoration methods such as super-resolution (SR) and denoising have become indispensable because they significantly outperform handcrafted algorithms. The approach proposed by Kim and others [33] was the first to introduce a deep network architecture based on residual learning for SR, which they called VDSR, and achieved a substantial boost in SR performance. The method proposed by Zhang and others [34] achieved an even greater improvement by exploiting residual dense blocks (RDBs), each of which consist of densely connected convolutional layers and a local skip connection. Kim and others [35] proposed a grouped residual dense network (GRDN), which extended earlier approaches by grouping multiple RDBs into grouped RDBs (GRDBs) and arranged multiple GRDBs in a network structure. They also incorporated a deeper architecture that allowed convolutional layers to process downsampled representations while adopting spatial and channel-wise attention layers. Based on this enhanced architecture, they obtained state-of-the-art (SOTA) performance on an image denoising task. Recently, Cho and others [36] utilized a GRDN [35] to reduce the artifacts caused by an image codec (namely, the intra-coding of VVC [24]) and achieved a noticeable quality improvement. However, the GRDN used by Cho and others [36] was optimized separately from the image codec.

### 3 | PROPOSED ARCHITECTURE

#### 3.1 | Cascading scheme for image compression and quality enhancement

Figure 1 presents our end-to-end joint learning scheme JointIQ-Net for performing image compression and

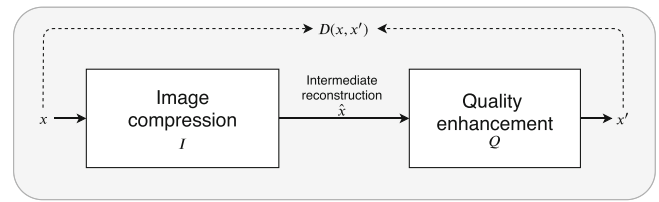


FIGURE 1 Joint learning scheme for image compression and quality enhancement.

quality enhancement in a cascade that provides high flexibility and extensibility. In particular, the proposed scheme can easily accommodate forthcoming advanced image quality enhancement networks and allows various combinations of image compression and quality enhancement methods. In other words, separately developed image compression networks and quality enhancement networks can be easily combined and jointly optimized in a unified architecture.

We compared two cascaded versions of a reference image compression model [6] and enhancement network. The first cascaded version was optimized for image compression and quality enhancement in an end-to-end manner. The reference image compression network and enhancement network were trained separately in the second cascaded version, where the enhancement network was trained on reconstructions outputted by the separately trained reference image compression network utilizing the same training dataset. The first end-to-end trained version outperformed the second, particularly in higher bit rate ranges (see Appendix A); therefore, we adopted end-to-end training of the cascaded model going forward. The quality enhancement network can be considered as the decoder component of the image compression network. Additionally, increasing the decoder complexity of the image compression network can result in a certain level of coding efficiency improvement. However, the goal of our study was to develop a flexible scheme for image compression and any image quality enhancement solution that can be independently developed separately from image compression. Furthermore, simply increasing the complexity of the decoder component may not provide a meaningful coding efficiency improvement due to its limited structure.

To select an appropriate quality enhancement network for our experiments, we combined a reference image compression method [6] with various quality improvement methods that are easy to incorporate and accessible, including VDSR [33], RDN [34], and GRDN [35], through cascaded connections. When the numbers of parameters and layers for the quality enhancement networks were adjusted to provide similar levels of computational complexity, GRDN [35] yielded

the best compression performance in combination with the image compression method. Therefore, we adopted GRDN [35] in our JointIQ-Net model. The corresponding test results are presented in Appendix A. It should be noted that although JointIQ-Net may be able to achieve even higher coding efficiency when combined with a more advanced enhancement network, we will leave this to future work because finding the best architecture for the enhancement network was not the main objective of this work.

The overall network architecture of JointIQ-Net is presented in Figure 2. As discussed in Section 3.1, our image compression network is connected to a GRDN model, which was adopted as a quality enhancement subnetwork, in a cascaded manner. The JointIQ-Net image compression network is based on an existing approach [6]. Therefore, we used the same rate-distortion (RD) optimization framework and transformation functions as the original study. JointIQ-Net transforms an input  $\mathbf{x}$  into latent representations  $\mathbf{y}$ , after which  $\mathbf{y}$  is quantized into  $\hat{\mathbf{y}}$ . Additionally, we use the hyperprior  $\hat{\mathbf{z}}$  proposed by Ballé and others [2], which further captures the spatial correlations of  $\hat{\mathbf{y}}$ . Accordingly, we use four fundamental transformation functions, namely, analysis transform  $g_a(\mathbf{x}; \phi_g)$ , synthesis transform  $g_s(\hat{\mathbf{y}}; \theta_g)$ , analysis transform  $h_a(\hat{\mathbf{y}}; \phi_h)$ , and synthesis transform  $h_s(\hat{\mathbf{z}}; \theta_h)$ , as described in previous works [2, 6]. Specifically, the analysis transform function  $g_a(\mathbf{x}; \phi_g)$  transforms an input image  $\mathbf{x}$  into a latent representation  $\mathbf{y}$  and the synthesis

transform  $g_s(\hat{\mathbf{y}}; \theta_g)$  reconstructs the image  $\hat{\mathbf{x}}$  from the quantized latent representation  $\hat{\mathbf{y}}$ . In contrast, the analysis transform  $h_a(\hat{\mathbf{y}}; \phi_h)$  captures the spatial redundancies of  $\hat{\mathbf{y}}$  in a latent representation  $\mathbf{z}$ , and the synthesis transform  $h_s(\hat{\mathbf{z}}; \theta_h)$  generates the context for the model estimation of  $\hat{\mathbf{y}}$ . We adopted the same architectures of  $g_a(\mathbf{x}; \phi_g)$ ,  $g_s(\hat{\mathbf{y}}; \theta_g)$ ,  $h_a(\hat{\mathbf{y}}; \phi_h)$ , and  $h_s(\hat{\mathbf{z}}; \theta_h)$  used in previous studies [2, 6].

The optimization process ensures that JointIQ-Net yields the lowest possible entropy values for  $\hat{\mathbf{y}}$  and  $\hat{\mathbf{z}}$  and that it yields a  $\hat{\mathbf{x}}$  reconstructed from  $\hat{\mathbf{y}}$  as similar to the original visual quality as possible. To achieve RD optimization, in addition to the distortion between the input  $\mathbf{x}$  and output  $\hat{\mathbf{x}}$ , the rate is calculated based on the distribution models of the latent representations for  $\hat{\mathbf{y}}$  and  $\hat{\mathbf{z}}$ . For the hyperprior  $\hat{\mathbf{z}}$ , we use a simple zero-mean Gaussian model convolved with  $\mathcal{U}(-1/2, 1/2)$ , whose standard deviation values are found during training, whereas the distribution parameters of the prior latent representation  $\hat{\mathbf{y}}$  are estimated by the model parameter estimator  $f$  in an autoregressive manner, as introduced in a previous method [6, 10]. However, we use the improved entropy model proposed in Section 4.

## 4 | IMPROVED ENTROPY MODEL

To address the limited ability of a single Gaussian-based entropy model, we adopted the GMM as a distribution

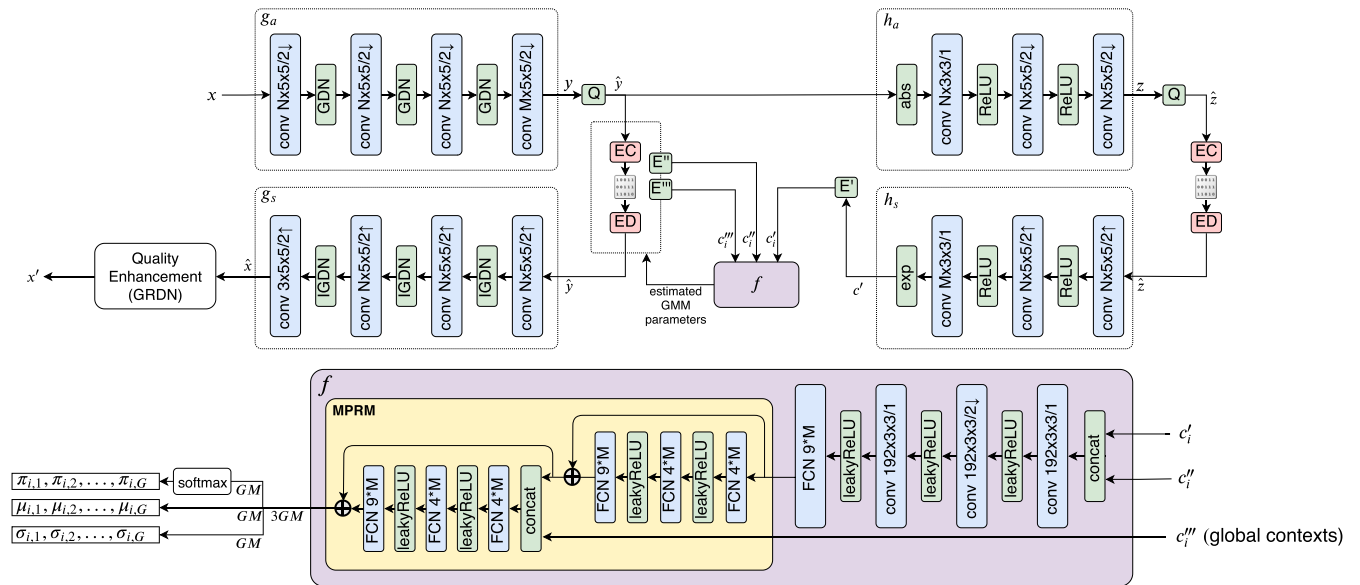


FIGURE 2 Architecture of JointIQ-Net. Each convolutional layer is represented as the number of filters  $\times$  the filter height  $\times$  the filter width divided by the downscale or upscale factor, where  $\uparrow$  and  $\downarrow$  denote the upscaling and downscaling via transposed convolutions, respectively. Input images are normalized to a scale between  $-1$  and  $1$ .  $N$  and  $M$  in the convolution layers denote the numbers of filters, whereas  $M$  in each fully connected layer denotes the number of nodes multiplied by the accompanying integer.

model for the latent representation  $y_i$ , which causes the estimated distribution to fit the actual distribution of each  $y_i$  more accurately. Accordingly,  $f$  in the proposed method estimates the parameters of the GMM, as described in Section 4.1. From a contextual perspective for autoregressive models, in addition to the two types of previously utilized contexts [6, 10], as well as the information reconstructed from the hyperprior  $\hat{\mathbf{z}}$  and adjacent known representations of  $\hat{\mathbf{y}}$ , we let  $f$  exploit the GC to estimate model parameters more precisely, as described in Section 4.2. In Figure 2, the three contexts (information from  $\hat{\mathbf{z}}$ , adjacent representations of  $\hat{\mathbf{y}}$ , and proposed GC) are denoted as  $\mathbf{c}'_i$ ,  $\mathbf{c}''_i$ , and  $\mathbf{c}'''_i$ , respectively, and the functions used to extract the three types of contexts are denoted as  $E'$ ,  $E''$ , and  $E'''$ , respectively.

Additionally, we enhance the structure of the model estimator  $f$  based on the proposal by Lee and others [6] by extending it to a new model estimator. The new model estimator incorporates a model parameter refinement module (MPRM) to improve the effectiveness of model parameter estimation, as shown in Figure 2. The MPRM utilizes a residual learning scheme that includes two residual blocks, each of which contains fully connected layers and corresponding nonlinear activation layers.

#### 4.1 | Joint optimization with a GMM-based entropy model

All elements proposed in our method are jointly optimized using the objective function defined in (1). Here, the RD optimization framework used is identical to that used in existing approaches [6, 10]. The objective function includes the rate and distortion terms, and the parameter  $\lambda$  is used to adjust the balance between the rate and distortion during the optimization process.

$$\mathcal{L} = R + \lambda D$$

$$\text{with } R = \mathbb{E}_{\mathbf{x} \sim p_x} \mathbb{E}_{\tilde{\mathbf{y}}, \tilde{\mathbf{z}} \sim q} \left[ -\log p_{\tilde{\mathbf{y}}|\tilde{\mathbf{z}}}(\tilde{\mathbf{y}}|\tilde{\mathbf{z}}) - \log p_{\tilde{\mathbf{z}}}(\tilde{\mathbf{z}}) \right]. \quad (1)$$

The rate term consists of the cross-entropies of  $\tilde{\mathbf{z}}$  and  $\tilde{\mathbf{y}}|\tilde{\mathbf{z}}$ . To handle the discontinuities caused by quantization, as in previous works [1, 2, 6, 10], a density function convolved with a uniform function  $\mathcal{U}(-1/2, 1/2)$  is used to approximate the PMF of  $\hat{y}_i$ . Correspondingly, for training, noisy representations  $\tilde{\mathbf{y}}$  and  $\tilde{\mathbf{z}}$  with uniform distributions and mean values of  $\mathbf{y}$  and  $\mathbf{z}$ , respectively, were used to fit the actual sample distributions to the PMF approximations. To model the distributions of  $\mathbf{z}$ , as in the aforementioned study [6], we used zero-mean Gaussian density functions whose standard deviations were optimized via training. However, we extended the entropy model for  $\tilde{\mathbf{y}}|\tilde{\mathbf{z}}$  based on the GMM as follows:

$$p_{\tilde{\mathbf{y}}|\tilde{\mathbf{z}}}(\tilde{\mathbf{y}}|\tilde{\mathbf{z}}) = \prod_i \left( \sum_{g=1}^G \pi_{i,g} \mathcal{N}(\mu_{i,g}, \sigma_{i,g}^2) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right) (\tilde{y}_i)$$

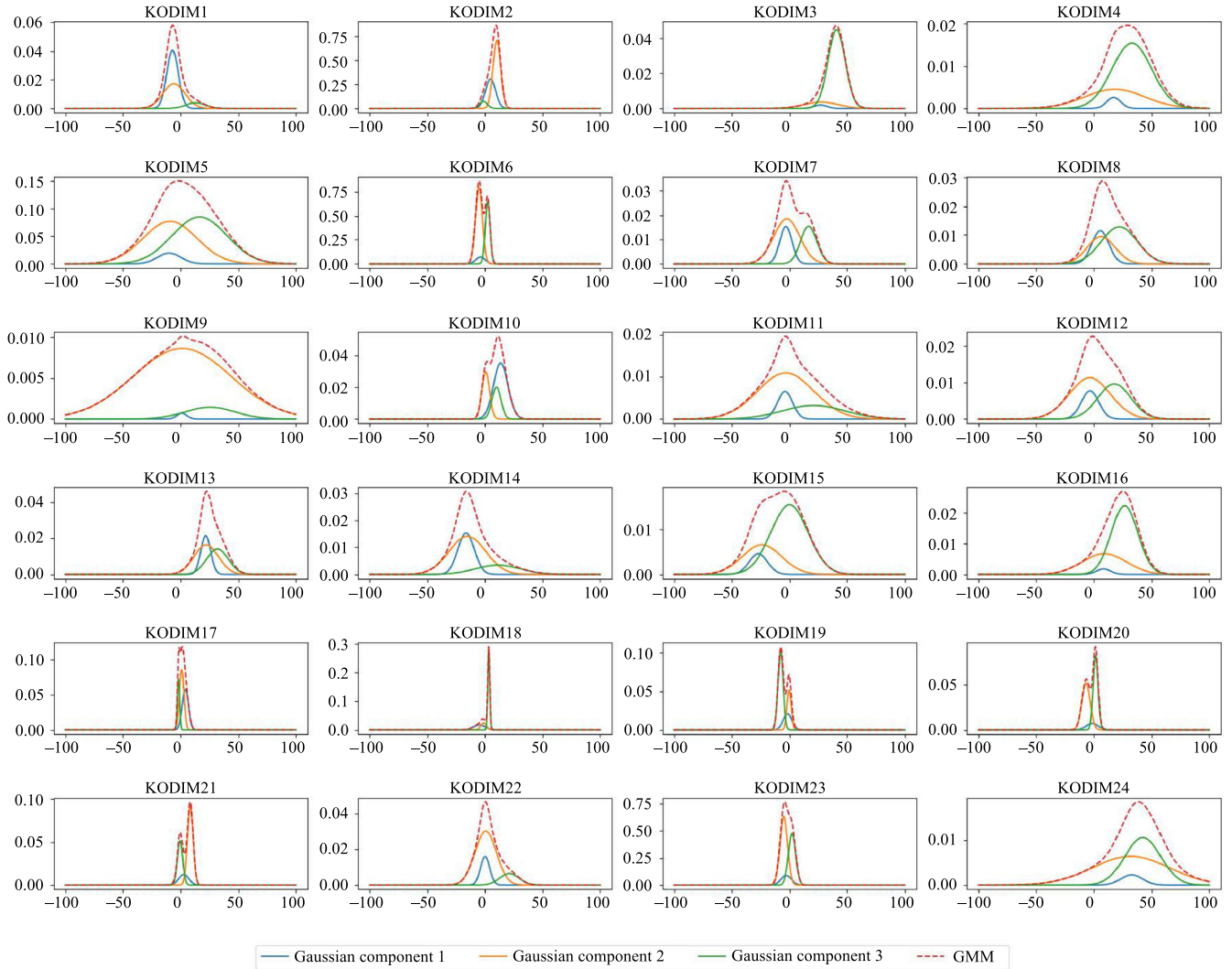
$$\text{with } \left\{ \pi_{i,g}, \mu_{i,g}, \sigma_{i,g} \mid 1 \leq g \leq G \right\} = f(\mathbf{c}'_i, \mathbf{c}''_i, \mathbf{c}'''_i), \quad (2)$$

where  $G$  is the number of Gaussian distribution functions. The distribution estimator  $f$  predicts  $3 \times G$  parameters such that each of the  $G$  Gaussian distributions has its own weight, mean, and standard deviation, which are denoted as  $\pi_{i,g}$ ,  $\mu_{i,g}$ , and  $\sigma_{i,g}$ , respectively. Figure 3 presents the estimated distributions of certain latent representations based on the estimated GMM parameters of our final model for the Kodak PhotoCD image dataset [37]. One can see that weighted sums of three Gaussian components are mostly skewed and difficult to model using single Gaussian functions. Furthermore, a few distributions (KODIM6, KODIM19, KODIM20, and KODIM21) had two peaks, although some of the peaks were very close to each other. It is noteworthy that the latent representations in Figure 3 were selected according to a predefined rule to visualize how GMMs assist with the generalization of the distributions for latent representations. However, in many cases, other latent representations have small values close to zero, resulting in values of zero following the quantization step. In Equation (1), the mean squared error (MSE) is used as the primary distortion term. We also present experimental results for MS-SSIM-optimized models [38]. Additional implementation details are provided in Appendix B.

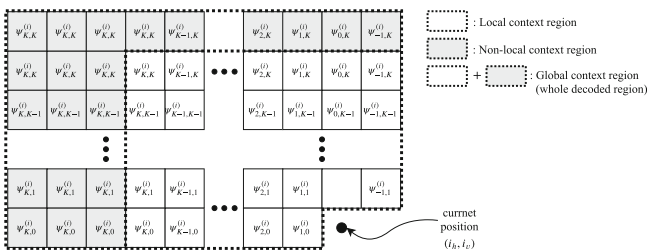
#### 4.2 | Extraction of GC

To obtain better contextual information for the current latent representation  $y_i$ , we can use a GC by aggregating all possible contexts from the entire area of the known representations  $\mathbf{y}_{<i}$  to estimate the distribution of  $y_i$ . To this end, we define the GC as information aggregated from the entire area of known representations  $\mathbf{y}_{<i}$ , which consists of local and nonlocal context regions. The local context region is within a fixed spatial distance, denoted as  $K$ , from the current representation  $y_i$  and the nonlocal region is the entire causal area outside the local context region. Figure 4 presents examples of local and nonlocal context regions.

For the GC  $\mathbf{c}'_i$ , we use a weighted mean value and weighted standard deviation value over the GC region of  $\tilde{\mathbf{y}}$ . We obtain the GC  $\mathbf{c}''_i$  from the  $\tilde{\mathbf{y}}$  representation, which is a latent representation linearly transformed from  $\tilde{\mathbf{y}}$  via a  $1 \times 1$  convolution layer, instead of from  $\hat{\mathbf{y}}$ , to capture the correlations across different channels of  $\hat{\mathbf{y}}$ .



**FIGURE 3** Distributions of several latent representations based on the estimated GMM parameters for the Kodak PhotoCD image dataset [37]. To demonstrate clearly how generally the proposed GMM model estimates the distributions of latent representations, we selected a representation  $\hat{y}_i$  for visualization for each image, according to the predefined rule  $\text{argmax}_i \pi_{i,0} \pi_{i,1} |\mu_{i,0} - \mu_{i,1}| + \pi_{i,1} \pi_{i,2} |\mu_{i,1} - \mu_{i,2}| + \pi_{i,2} \pi_{i,0} |\mu_{i,2} - \mu_{i,0}|$ . One can see that the estimated GMM models have various shapes that cannot be expressed by a single Gaussian model. To obtain the PMF of the discrete representation  $\hat{y}_i$  from a continuous GMM density function, we simply convolve the GMM density function with a uniform function  $\mathcal{U}(-1/2, 1/2)$ , as done in earlier studies [1, 2, 6, 10]. In this implementation, the probability mass function (PMF) (or CDF) of the GMM is simply calculated via the weighted summation of separate PMFs (or CDFs) from constitutive Gaussian components.



**FIGURE 4** Example  $\mathbf{a}^{(i)}$  with a set of  $\psi^{(i_c)}$  variables mapped to the global context region. The softmax operation is applied to  $\mathbf{a}^{(i)}$  to determine the normalized weight  $\mathbf{w}^{(i)}$

Specifically, the GC  $\mathbf{c}_i^{(i)} = \{\mu_i^*, \sigma_i^*\}$  consists of the weighted mean  $\mu_i^*$  and weighted standard deviation  $\sigma_i^*$ , which are defined as follows:

$$\mu_i^* = \sum_{k,l \in S} w_{k,l}^{(i)} \hat{y}_{i_h-k, i_w-l}^{(i_c)} \quad (3)$$

$$\sigma_i^* = \sqrt{\frac{\sum_{k,l \in S} w_{k,l}^{(i)} (\hat{y}_{i_h-k, i_w-l}^{(i_c)} - \mu_i^*)^2}{1 - \sum_{k,l \in S} w_{k,l}^{(i)}}} \quad (4)$$

where  $\mathbf{i} = [i_c, i_h, i_v]$  is a three-dimensional spatio-channel-wise position index that indicates the current position  $(i_h, i_v)$  in the  $i_c$ th channel.  $w_{k,l}^{(i)}$  is a weight variable for the relative coordinates  $(k, l)$  based on the current location  $(i_h, i_v)$  and  $\dot{\mathbf{y}}_{i_h-k, i_v-l}^{(i)}$  is a representation of  $\dot{\mathbf{y}}^{(i)}$  at a location  $(i_h-k, i_v-l)$  within the GC region  $S$ .  $\dot{\mathbf{y}}^{(i_c)}$  denote the two-dimensional representations within the  $i_c$ th channel of  $\dot{\mathbf{y}}$ . The weight variables in  $\mathbf{w}^{(i)}$  are normalized weights multiplied element wise by  $\dot{\mathbf{y}}^{(i_c)}$  for the weighted mean in (3) and by the difference squares of  $(\dot{\mathbf{y}}_{i_h-k, i_v-l}^{(i_c)} - \mu_{\mathbf{i}}^*)$  in Equation (4). The denominator in (4) ensures an unbiased estimation.

The key issue is to find an optimal set of weight variables  $\mathbf{w}^{(i)}$  at every location  $\mathbf{i}$  in a fixed number of trainable variables  $\boldsymbol{\psi}^{(i_c)}$ . The trainable variables  $\boldsymbol{\psi}^{(i_c)}$  are used as weight variables (before softmax normalization), but they cover only the local context region, meaning the area within a fixed distance  $K$ , as shown in Figure 4. To ensure that the weight variables cover the entire area of the GC region, we expand  $\boldsymbol{\psi}^{(i_c)}$  by allocating the nearest  $\boldsymbol{\psi}^{(i_c)}$  values to the nonlocal context regions, as shown in Figure 4. Consequently, we obtain a set of expanded  $\boldsymbol{\psi}^{(i_c)}$  variables denoted as  $\boldsymbol{\alpha}^{(i)}$ , which correspond to the GC region. Subsequently, the final weight variables  $\mathbf{w}^{(i)}$  are calculated by normalizing  $\boldsymbol{\alpha}^{(i)}$  via a softmax operation as follows:

$$\mathbf{w}^{(i)} = \text{softmax}(\boldsymbol{\alpha}^{(i)}), \quad (5)$$

where  $\boldsymbol{\alpha}^{(i)} = \left\{ \boldsymbol{\psi}_{\text{clip}(k,K), \text{clip}(l,K)}^{(i_c)} \mid k, l \in S \right\}$  and  $\text{clip}(x, K) = \max(-K, \min(K, x))$ . It should be noted that  $\boldsymbol{\psi}_{k,l}^{(i_c)}$

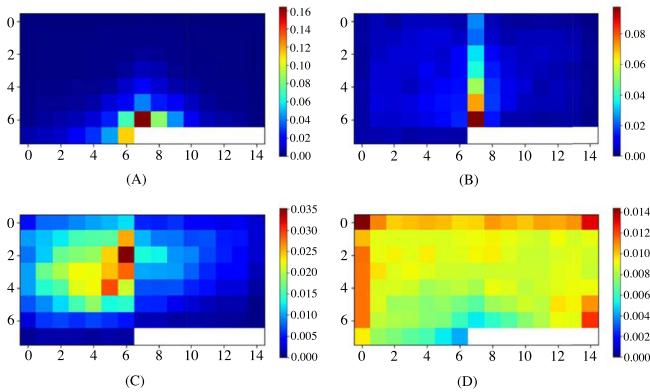


FIGURE 5 Examples of the trained  $\boldsymbol{\psi}^{(i_c)}$ , each of which is a set of weights for spatially aggregating contextual information from the entire spatial area of a specific channel of  $\dot{\mathbf{y}}$ . When a particular spatial position of  $\dot{\mathbf{y}}$  is not covered by  $\boldsymbol{\psi}^{(i_c)}$  due to the limited size of  $\boldsymbol{\psi}^{(i_c)}$ , the nearest weight value in  $\boldsymbol{\psi}^{(i_c)}$  is shared with that spatial position.

remains unchanged in the same channel. Figure 5 presents the trained  $\boldsymbol{\psi}^{(i_c)}$  examples for several channels of  $\dot{\mathbf{y}}$ . Figure 5A presents a case in which the context of the channel depends on neighboring representations directly adjacent to the current latent representation, whereas Figure 5D presents a case in which the context of the channel depends on broadly spread neighboring representations.

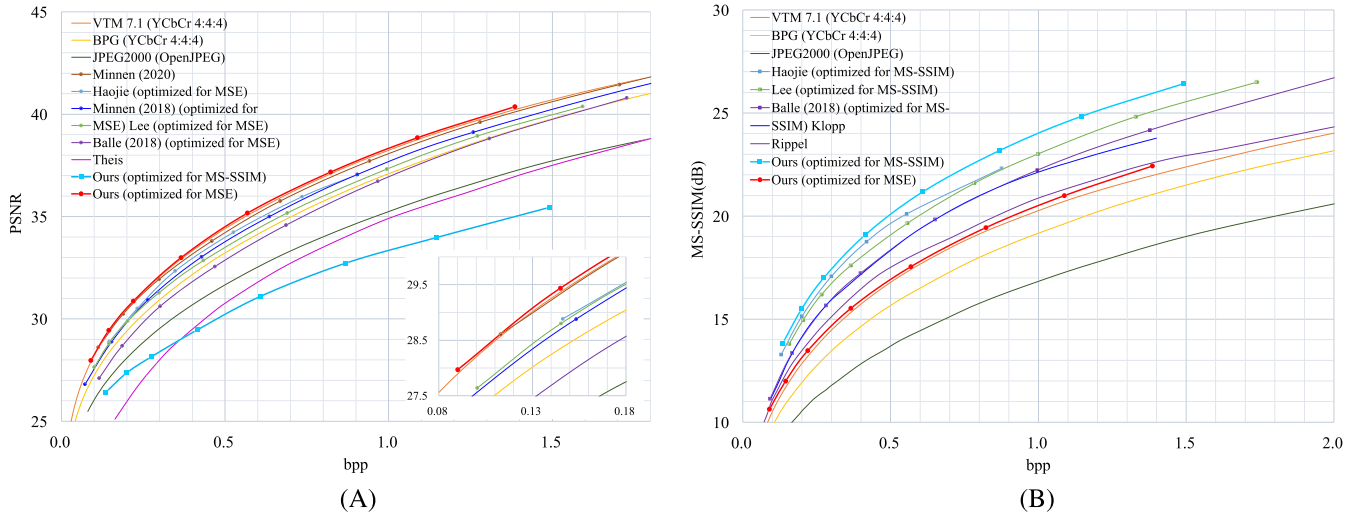
## 5 | EXPERIMENTS

### 5.1 | Experimental setup

To verify the performance of the proposed method, we measured the average bits per pixel (bpp) and quality of reconstructed images for the Kodak PhotoCD [37] (24 images of  $768 \times 512$  or  $512 \times 768$  pixels in size), CLIC validation [39] (102 images ranging in size from  $512 \times 384$  to  $1520 \times 2048$  pixels), and Tecnick [40] (40 images of  $1200 \times 1200$  pixels in size) image datasets, which are commonly used in the image compression research area. The PSNR and MS-SSIM metrics were used to measure the quality of the results. PSNR is a traditional MSE-based metric, whereas MS-SSIM [38] is a more recent human-perception-oriented metric. Both the PSNR and MS-SSIM metrics have been widely used in recent image and video compression studies. For each quality metric, eight models were trained with different  $\lambda$  values, after which they were evaluated by comparing the resulting RD curves to those of existing ANN-based approaches [2, 5, 6, 10, 11, 14, 31] and the conventional codecs of VVC intra-coding (VTM 7.1 [23]), BPG [19], and JPEG2000 [20]. It should be noted that in the case of VVC intra-coding (VTM 7.1 [23]) and BPG [19], we used the YCbCr 4:4:4 format because it yields the best results in terms of RGB PSNR BD rate. The input RGB images were converted into YCbCr 4:4:4 format before compression and the reconstruction results were converted back into the RGB format. We compared the results in the range of 0.1 to 1.5 bpp.

### 5.2 | Experimental results

We compared the compression performance of our method to that of the aforementioned approaches based on RD curves in terms of PSNR and MS-SSIM. Figure 6 presents the coding efficiency curves for JointIQ-Net, VVC intra-coding (VTM 7.1 [23]), BPG [19], JPEG2000 [20], and the deep learning-based SOTA methods [2, 5, 6, 10, 11, 14, 31] on the Kodak PhotoCD image dataset [37]. As shown in Figure 6A,B, JointIQ-Net



**FIGURE 6** RD curves of the proposed method and compared methods for the Kodak PhotoCD image dataset [37]. The left and right plots represent RD curves in terms of (A) PSNR and (B) MS-SSIM, respectively. Note that the measured MS-SSIM values are presented in units of decibels, as in previous works [2,6,10], to clarify performance differences.

**TABLE 1** PSNR BD-rate gains of JointIQ-Net (MSE optimization) against existing methods on the Kodak PhotoCD image dataset [37].

Reference method	BD-rate gain (%)
VVC intra-coding (YCbCr 4:4:4) [23]	1.65
BPG (YCbCr 4:4:4) [19]	22.57
JPEG2000 [20]	45.48
Minnen and Singh [11]	4.07
Liu et al. (opt. for MSE) [31]	10.94
Minnen et al. [10]	13.17
Lee (opt. for MSE) [6]	16.96
Ballé et al. (opt. for MSE) [2]	26.58
Theis et al. [15]	50.73

outperformed all of the image compression methods in terms of both PSNR and MS-SSIM. The BD-rate gains of JointIQ-Net compared with VVC intra-coding (VTM 7.1 [23]), BPG [19], JPEG2000 [20], and the deep learning-based SOTA methods [2, 5, 6, 10, 11, 14, 31] are summarized in Tables 1 and 2. It should be noted that JointIQ-Net is the *first* deep image compression method that surpasses VVC intra-coding in terms of the PSNR BD rate. In Appendix C, we present the test results for the CLIC [39] and Tecnick [40] image sets, demonstrating that the proposed method significantly outperformed VVC intra-coding (VTM 7.1 [23]) by 4.85% and 7.12%, respectively, in terms of the PSNR BD rate.

Figure 7 presents a visual comparison of several reconstructed images produced by JointIQ-Net and the existing methods [6, 19, 23]. At similar compression

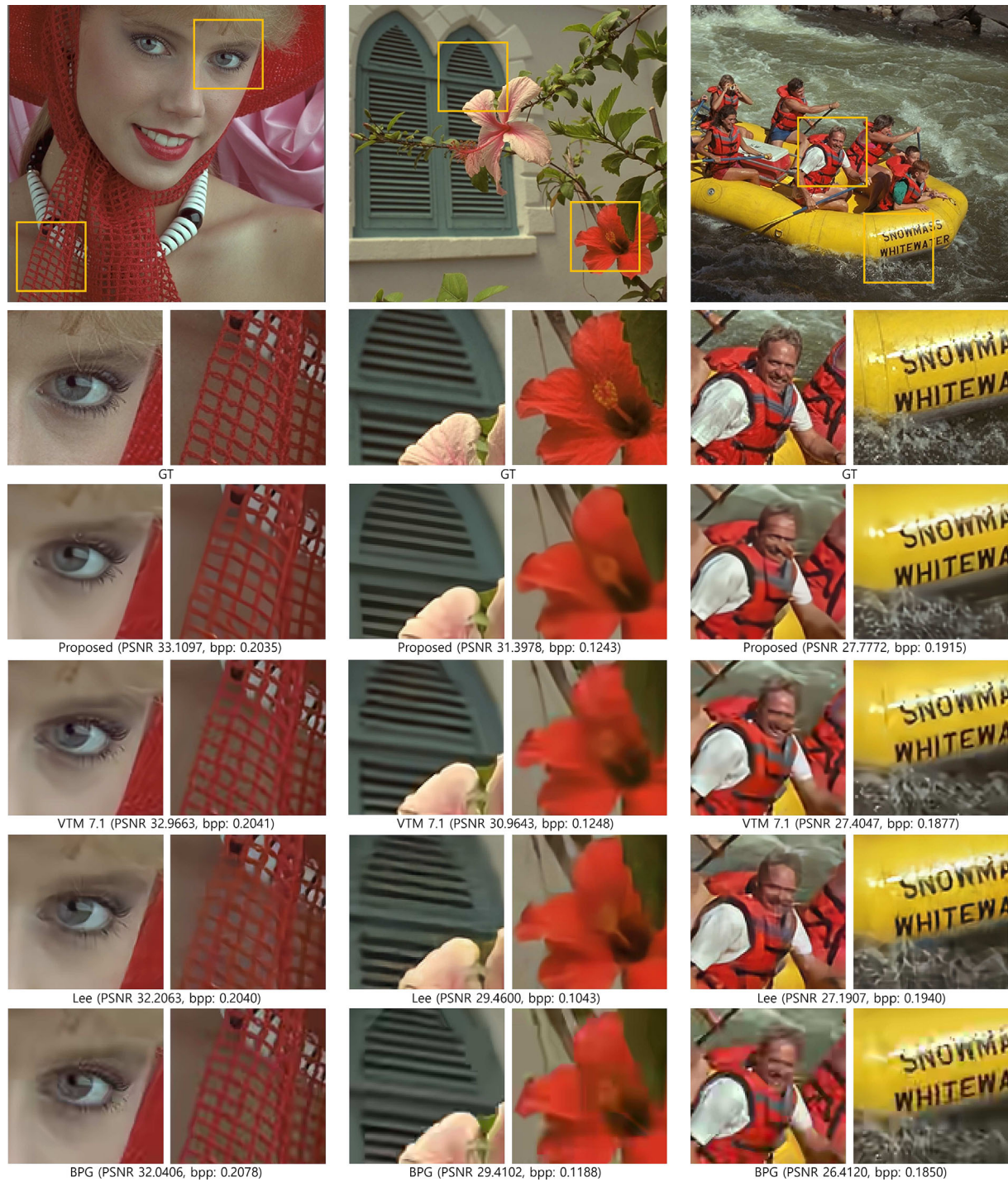
**TABLE 2** MS-SSIM BD-rate gains of JointIQ-Net (MS-SSIM optimization) against existing methods on the Kodak PhotoCD image dataset [37].

Reference method	BD-rate gain (%)
VVC intra-coding (YCbCr 4:4:4) [23]	48.40
BPG (YCbCr 4:4:4) [19]	57.35
JPEG2000 [20]	73.65
Liu et al. (opt. for MS-SSIM) [31]	9.07
Lee et al. (opt. for MS-SSIM) [6]	14.83
Ballé et al. (opt. for MS-SSIM) [2]	26.65
Klopp et al. [5]	28.35
Rippel and Bourdev [14]	40.47

ratios, as shown in Figure 7, JointIQ-Net yielded reconstructed images of higher quality in terms of both qualitative viewing and quantitative measures (PSNR).

### 5.3 | Ablation study

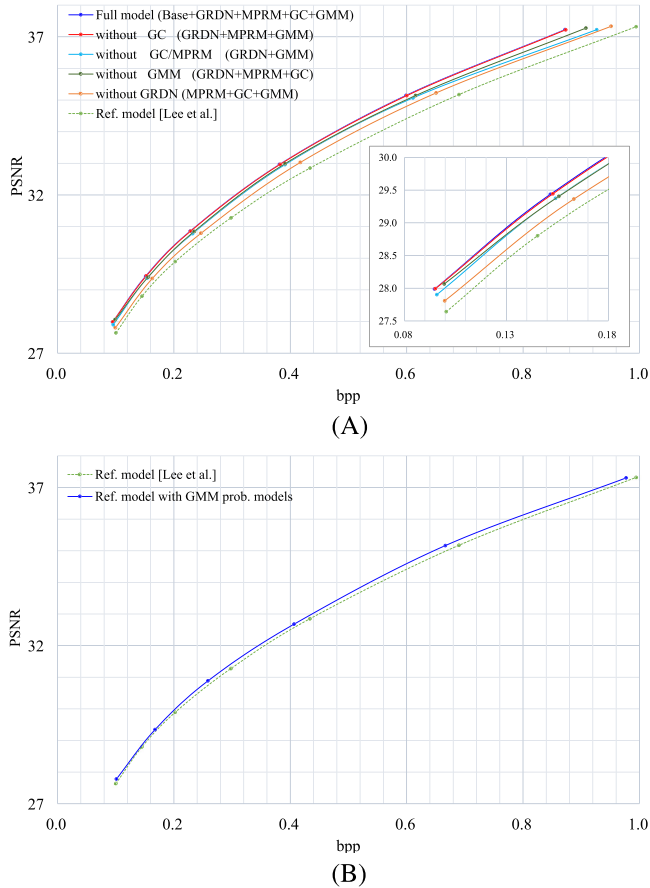
To verify the effectiveness of each proposed element, we conducted an ablation study in which we excluded each proposed element from the full model and trained the models in the same manner as described in Section 5. Subsequently, we compared the test results of each model with those of the full model. Four different models were evaluated: one each with the GRDN [35] subnetwork excluded, GMM excluded, MPRM excluded, and GC excluded. The GC was also excluded when excluding the MPRM because the GC is processed by the MPRM in our



**FIGURE 7** Comparison of sample reconstruction images showing the ground truth, proposed JointIQ-Net results, VVC intra-coding results (VTM 7.1) [23], results of the method presented by Lee et al. [6], and BPG results [19].

full model. Additionally, as a baseline, the compression performance of the model presented by Lee and others [6] was included in the comparison. Figure 8A presents the PSNR performances of the various versions of the model considered in the ablation study. One can see that when the GRDN [35] is excluded, a significant performance degradation occurs. This indicates that the proposed joint

learning scheme plays an important role in improving coding efficiency. When a single Gaussian model is used instead of a GMM, a similar level of performance degradation occurs over the entire bpp range. To assess the effectiveness of the proposed GMM model, we performed an additional comparison test between the reference model using a single Gaussian model and the same

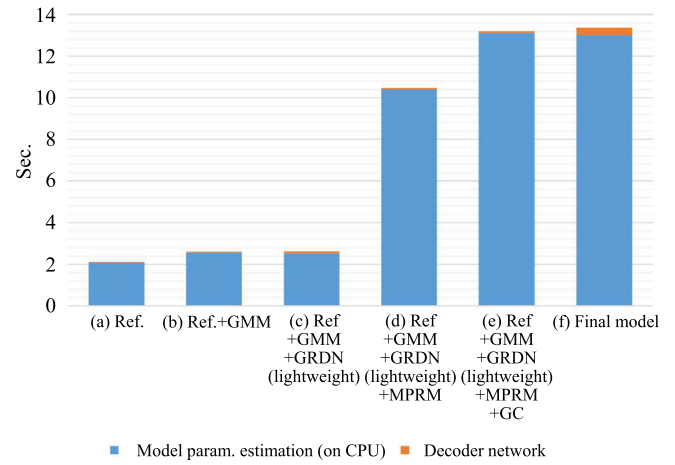


**FIGURE 8** PSNR performance on the Kodak PhotoCD [37] image set: (A) JointIQ-Net variations and (B) a reference model [6] using a single Gaussian model and the same model with the proposed GMM model.

model with the proposed GMM model, without other architectural or contextual changes. As shown in Figure 8B, we obtained a coding gain of 3.63% when using the proposed GMM model compared with the reference model. In this case, the two networks had the same architecture, except for the number of output nodes of the model estimator  $f$ . The GC also improved performance, but only yielded a relatively marginal gain of 0.33%. To verify the effectiveness of the GC, we compared our final model in Figure 6 to its fine-tuned version, which was trained for a further 0.5M iterations with dummy GC values set to zero. As a result, we obtained a similar 0.27% improvement, indicating that the proposed GC exploitation helps improve coding efficiency. In the future, we plan to study how to utilize global representations more effectively. When both the MPRM and GC were excluded, performance degradation became more noticeable. The results also indicate that the MPRM has a greater impact in the higher bpp range. Table 3 compares the quantitative results of the final model and each of the element-excluded models in terms of BD-rate loss.

**TABLE 3** BD-rate losses of element-excluded models in comparison with the full model, namely, JointIQ-Net (anchor).

Models	BD-rate loss (%)
Without GRDN	8.87
Without GC	0.33
Without GC/MPRM	3.96
Without GMM	3.01



**FIGURE 9** Decoding time of our methods: we measured the average decoding time of each model on the Kodak PhotoCD image dataset [37] with an RTX Titan GPU and Intel Xeon Gold 5122 CPU.

## 5.4 | Decoding time

To verify how each component of JointIQ-Net affects decoder complexity, we measured the average decoding times for the JointIQ-Net variations on the Kodak PhotoCD image dataset [37]. It should be noted that the functions related to the model parameters, namely,  $f$  (including the MPRM) and  $h_s$ , must operate identically as on the encoder side without precision differences in the floating point computations between the encoders and decoders, but GPUs sometimes introduce these types of discrepancies, as discussed in the literature [41]. Accordingly, we used a CPU for the model parameter-related functions  $f$  and  $h_s$ . The synthesis transform function  $g_s$  and quality enhancement subnetwork  $Q$  are free from this computational discrepancy issue between the encoder and decoder, so they were executed on a GPU.

Figure 9 presents the decoding times of our models ( $N$ : 192,  $M$ : 256) for a moderate bpp range, where the proposed components are added incrementally. Entropy model parameter estimation requires much more time than the synthesis transform function  $g_s$  and quality enhancement subnetwork  $Q$  on the GPU because entropy

model parameter estimation is processed in an autoregressive manner on a CPU. This slow operation can be accelerated using wavefront parallel processing, which is beyond the scope of this study because our goal was simply to compare the numbers of operations required by the components of JointIQ-Net. Additionally, the proposed GMM requires  $G$  times the computational complexity to determine the CDF of each quantized latent representation, which does not significantly increase decoding time because  $G$  elementary CDFs can be calculated in parallel. Therefore, for a more precise comparison, we excluded the time required for CDF computation and symbol (quantized latent representation) extraction, as shown in Figure 9. It should be noted that the GMM in Figure 9B and image quality enhancement network in Figure 9C increase decoding time by a negligible amount, indicating that these two components are effective and simple enough to be adopted in real-world applications. Furthermore, it should be noted that the GMM model and quality enhancement subnetwork can be exploited on top of hyperprior-based approaches [2], where decoding can be performed fully in parallel. However, the MPRM requires a larger portion of the total decoding time as a result of its CPU operations, which also results in a significant coding gain. We plan to address this issue in our future studies.

## 6 | CONCLUSION

In this paper, we proposed a novel image compression method called JointIQ-Net that outperforms VVC intra-coding (VTM 7.1), BPG, JPEG2000, and SOTA ANN-based image compression approaches. To the best of our knowledge, the proposed JointIQ-Net is the *first* learned image compression approach that surpasses VVC intra-coding in terms of both PSNR and MS-SSIM. To improve the coding efficiency of JointIQ-Net, we designed a novel joint learning scheme that incorporates both image compression and quality enhancement with an improved entropy model that utilizes a GMM as a more generalized distribution model for transformed latent representations. Additionally, we enhanced model parameter estimation by utilizing GCs, which can reduce the remaining correlations over the global region of transformed representations.

## ACKNOWLEDGMENTS

This work was supported by Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2017-0-00072, Development of Audio/Video Coding and Light Field Media Fundamental Technologies for Ultra Realistic Tera-media).

## CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest.

## ORCID

Munchurl Kim  <https://orcid.org/0000-0003-1634-7722>

## REFERENCES

1. J. Ballé, V. Laparra, and E. P. Simoncelli, *End-to-end optimized image compression*, (5th Int. Conf. on Learning Representations, Toulon, France), 2017, DOI [10.48550/arXiv.1611.01704](https://doi.org/10.48550/arXiv.1611.01704).
2. J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, *Variational image compression with a scale hyperprior*, (6th Int. Conf. on Learning Representations, Vancouver, Canada), 2018, DOI [10.48550/arXiv.1802.01436](https://doi.org/10.48550/arXiv.1802.01436).
3. D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, *Checkerboard context model for efficient learned image compression*, (Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit., Nashville, TN, USA), 2021, pp. 14771–14780.
4. N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. Jin Hwang, J. Shor, and G. Toderici, *Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks*, (IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Salt Lake City, UT, USA), 2018, DOI [10.1109/CVPR.2018.00461](https://doi.org/10.1109/CVPR.2018.00461).
5. J. P. Klopp, Y.-C. F. Wang, S.-Y. Chien, and L.-G. Chen, *Learning a code-space predictor by exploiting intra-image-dependencies*, (British Mach. Vision Conf., Newcastle Upon Tyne, UK), 2018. <https://bmva-archive.org.uk/bmvc/2018/contents/papers/0491.pdf>.
6. J. Lee, S. Cho, and S.-K. Beack, *Context-adaptive entropy model for end-to-end optimized image compression*, (7th Int. Conf. Learn. Representations, New Orleans, LA, USA), 2019.
7. M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, *Learning convolutional networks for content-weighted image compression*, (Proc. IEEE Conf. Comput. Vision Pattern Recognit., Salt Lake City, UT, USA), 2018, pp. 3214–3223.
8. G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, *DVC: an end-to-end deep video compression framework*, arXiv preprint, 2018, DOI [10.48550/arXiv.1812.00101](https://doi.org/10.48550/arXiv.1812.00101).
9. F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, *Conditional probability models for deep image compression*, (Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Salt Lake City, UT, USA), 2018, DOI [10.1109/CVPR.2018.00462](https://doi.org/10.1109/CVPR.2018.00462).
10. D. Minnen, J. Ballé, and G. Toderici, *Joint autoregressive and hierarchical priors for learned image compression*, 32nd Conference on Neural information processing Systems, Montreal, Canada), 2018.
11. D. Minnen and S. Singh, *Channel-wise autoregressive entropy models for learned image compression*, (IEEE Int. Conf. Image Process. (ICIP), Abu Dhabi, United Arab Emirates), 2020, DOI [10.1109/ICIP40778.2020.9190935](https://doi.org/10.1109/ICIP40778.2020.9190935).
12. W.-S. Park and M. Kim, *Deep predictive video compression with bi-directional prediction*, arXiv preprint, 2019, DOI [10.48550/arXiv.1904.02909](https://doi.org/10.48550/arXiv.1904.02909).
13. Y. Qian, M. Lin, X. Sun, Z. Tan, and R. Jin, *Entroformer: a transformer-based entropy model for learned image compression*, arXiv preprint, 2022, DOI [10.48550/arXiv.2202.05492](https://doi.org/10.48550/arXiv.2202.05492).
14. O. Rippel and L. Bourdev, *Real-time adaptive image compression*, (34th International Conference on Machine Learning, Sydney, Australia), 2017, DOI [10.48550/arXiv.1705.05823](https://doi.org/10.48550/arXiv.1705.05823).

15. L. Theis, W. Shi, A. Cunningham, and F. Huszr, *Lossy image compression with compressive autoencoders*, 5th International Conference on Learning Representations, Toulon, France), 2017, DOI [10.48550/arXiv.1703.00395](https://doi.org/10.48550/arXiv.1703.00395).
16. G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, *Variable rate image compression with recurrent neural networks*, (4th Int. Conf. Learn. Representations, ICLR 2016, Conference Track Proceedings, San Juan, Puerto Rico), 2016, DOI [10.48550/arXiv.1511.06085](https://doi.org/10.48550/arXiv.1511.06085).
17. G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, *Full resolution image compression with recurrent neural networks*, (IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Honolulu, HI, USA), 2017, DOI [10.1109/CVPR.2017.577](https://doi.org/10.1109/CVPR.2017.577).
18. Y. Zhu, Y. Yang, and T. Cohen, *Transformer-based transform coding*, (Int. Conf. Learn. Representations), 2021.
19. F. Bellard, *BPG image format*, 2014. <http://bellard.org/bpg/>.
20. D. S. Taubman and M. W. Marcellin, *JPEG2000: image compression fundamentals, standards and practice*, Kluwer Academic Publishers, Norwell, MA, USA, 2001.
21. D. Glasner, S. Bagon, and M. Irani, *Super-resolution from a single image*, (IEEE Int. Conf. Comput. Vision (ICCV), Kyoto, Japan), 2009, pp. 349–356.
22. D. Y. Lee, J. Lee, J.-H. Choi, J. Kim, H. Y. Kim, and J. S. Choi, *GPU-based real-time super-resolution system for high-quality UHD video up-conversion*, *J. Supercomput.* **74** (2018), no. 3, 456–484.
23. Versatile video coding reference software version 7.1 (VTM-7.1), 2019, [https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware\\_VTM/tags/VTM-7.1](https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-7.1).
24. S. L. Bross: Versatile video coding (Draft 5). JVET (ISO/IEC, Geneva, Switzerland), 2019. Draft.
25. N. Brahim, T. Bouden, T. Brahim, and L. Boubchir, *A novel and efficient 8-point DCT approximation for image compression*, *Multimed. Tools Appl.* **79** (2020), no. 11–12, 7615–7631.
26. D. Bscones, C. Gonzalez, and D. Mozos, *Hyperspectral image compression using vector quantization, PCA and JPEG2000*, *Remote Sensing* **10** (2018), no. 6, 907.
27. S. Kim and J. Kang, *Voxel-wise UV parameterization and view-dependent texture synthesis for immersive rendering of truncated signed distance field scene model*, *ETRI J.* **44** (2022), no. 1, 51–61.
28. W. Xiao, N. Wan, A. Hong, and X. Chen, *A fast JPEG image compression algorithm based on DCT*, (IEEE Int. Conf. Smart Cloud (SMARTCLOUD), Washington, DC, USA), 2020, pp. 106–110.
29. R. J. Yadav and M. S. Nagmode, *Compression of hyperspectral image using PCA-DCT technology*, *Innovations in electronics and communication engineering*, H. S. Saini, R. K. Singh, and K. S. Reddy, (eds.), Springer Singapore, Singapore, 2018, pp. 269–277.
30. Information technology—high efficiency coding and media delivery in heterogeneous environments—part 2: high efficiency video coding, (ISO/IEC, Geneva, Switzerland), 2013. Standard.
31. H. Liu, T. Chen, P. Guo, Q. Shen, X. Cao, Y. Wang, and Z. Ma, *Non-local attention optimized deep image compression*, arXiv preprint, 2019, DOI [10.48550/arXiv.1904.09757](https://doi.org/10.48550/arXiv.1904.09757).
32. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, *Swin transformer: hierarchical vision transformer using shifted windows*, (Proc. IEEE/CVF Int. Conf. Comput. Vision, Montreal, Canada), 2021, pp. 10012–10022.
33. J. Kim, J. K. Lee, and K. M. Lee, *Accurate image super-resolution using very deep convolutional networks*, (IEEE Conf. Comput. Vision Pattern Recognit. (CVPR ORAL), Las Vegas, NV, USA), 2016, DOI [10.1109/CVPR.2016.182](https://doi.org/10.1109/CVPR.2016.182).
34. Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, *Residual dense network for image super-resolution*, (IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Salt Lake City, UT, USA), 2018, DOI [10.1109/CVPR.2018.00262](https://doi.org/10.1109/CVPR.2018.00262).
35. D.-W. Kim, J. R. Chung, and S.-W. Jung, *GRDN: grouped residual dense network for real image denoising and DAN-based real-world noise modeling*, (IEEE Conf. Comput. Vision Pattern Recognit. (CVPR) Workshops, Long Beach, CA, USA), 2019, DOI [10.1109/CVPRW.2019.00261](https://doi.org/10.1109/CVPRW.2019.00261).
36. S. Cho, J. Lee, J. Kim, and Y. Kim, *Low bit-rate image compression based on post-processing with grouped residual dense network*, (IEEE Conf. Comput. Vision Pattern Recognit. (CVPR) Workshops, Long Beach, CA, USA), 2019.
37. E. Kodak, *Kodak lossless true color image suite (PhotoCD PCD0992)*, 1993, <http://r0k.us/graphics/kodak/>.
38. Z. Wang, E. P. Simoncelli, and A. C. Bovik, *Multiscale structural similarity for image quality assessment*, (The Thirty-Seventh Asilomar Conf. Signals Syst. Comput., Pacific Grove, CA, USA), 2003, DOI [10.1109/ACSSC.2003.1292216](https://doi.org/10.1109/ACSSC.2003.1292216).
39. Workshop and challenge on learned image compression, 2019, <https://www.compression.cc/>.
40. N. Asuni and A. Giachetti, *TESTIMAGES: a large-scale archive for testing visual devices and basic image processing algorithms*, (Smart Tools and Apps for Graphics, Cagliari, Italy), 2014, DOI [10.2312/stag.20141242](https://doi.org/10.2312/stag.20141242).
41. J. Ballé, N. Johnston, and D. Minnen, *Integer network for data compression with latent-variable models*, (7th Int. Conf. Learn. Representations, New Orleans, LA, USA), 2019. <https://openreview.net/forum?id=SIzz2i0cY7>.
42. D. P. Kingma and J. Ba, *Adam: a method for stochastic optimization*, (3rd Int. Conf. Learn. Representations, San Diego, CA, USA), 2015, DOI [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980).

## AUTHOR BIOGRAPHIES



**Jooyoung Lee** received the BS degree in Multimedia from Ajou University, Suwon, South Korea, in 2003, and the MS degree in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2006.

He is currently pursuing the PhD degree in Korea Advanced Institute of Science and Technology (KAIST). Since 2006, he has worked for Electronics and Telecommunications Research Institute, Daejeon, Korea, as a senior research engineer. He has been involved in standardization activities of 3D broadcasting systems in Advanced Television Systems Committee (ATSC). His current research interests include learned image and video compression, deep learning for image restoration, visual quality enhancement, and perceptual video coding.



**Seunghyun Cho** received the BS degree in Electrical Engineering from Kyungpook National University, Daegu, Korea, in 2003, and the MS and PhD degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2006 and 2015, respectively. He worked for Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, as a principal researcher from 2006 to 2019 and for Kyungnam University, Gyeongsangnam-do, Korea, as an assistant professor from 2020 to 2021. Currently, he is working for Agency for Defense Development (ADD), Daejeon, Korea, as a Senior Researcher. His research interests include embedded systems, video coding algorithms, and deep learning-based signal processing.

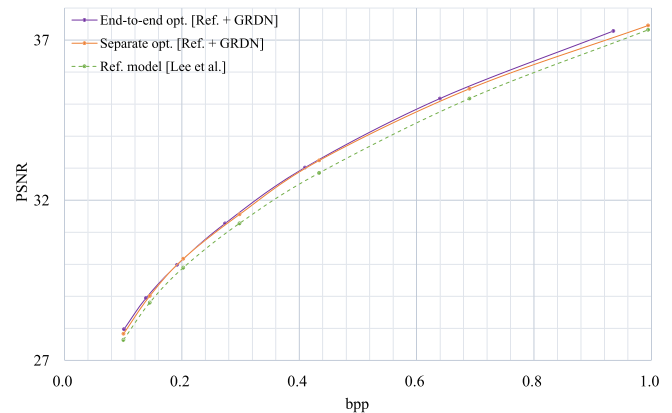


**Munchurl Kim** received the BE degree in Electronics from Kyungpook National University, Daegu, South Korea, in 1989, and the ME and PhD degrees in Electrical and Computer Engineering from the University of Florida, Gainesville, in 1992 and 1996, respectively. After his graduation, he joined the Electronics and Telecommunications Research Institute, Daejeon, Korea, as a senior research staff member, where he led the Realistic Broadcasting Media Research Team. In 2001, he was an assistant professor with the School of Engineering, Information and Communications University (ICU), Daejeon. Since 2009, he has been with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, where he is now a full professor. He had been involved with scalable video coding and high efficiency video coding (HEVC) in JCT-VC standardization activities of ITU-T VCEG and ISO/IEC MPEG. His current research interests include deep learning for image restoration and visual quality enhancement, deep video compression, perceptual video coding, visual quality assessments, computational photography, machine learning, and pattern recognition.

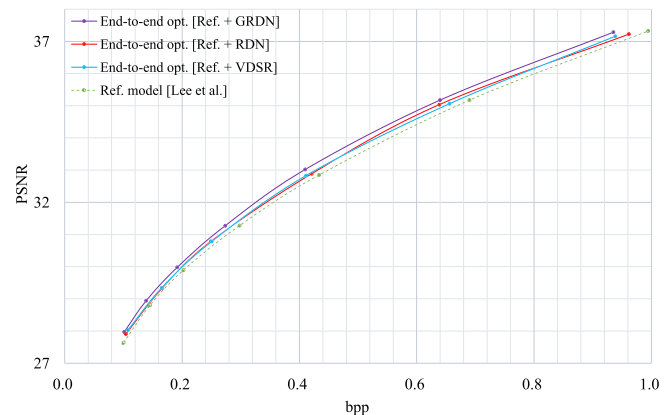
**How to cite this article:** J. Lee, S. Cho, and M. Kim, *An end-to-end joint learning scheme of image compression and quality enhancement with improved entropy minimization*, ETRI Journal **46** (2024), 935–949, DOI [10.4218/etrij.2023-0275](https://doi.org/10.4218/etrij.2023-0275).

## APPENDIX A: ADDITIONAL EXPERIMENTAL RESULTS FOR CHOOSING AN ENHANCEMENT NETWORK

Figure A1 presents the coding efficiencies of the two JointIQ-Net variations for the different training methods. We jointly trained the image compression and enhancement subnetworks in one version and separately trained them in the other version. Figure A2 presents the coding efficiencies of the JointIQ-Net variations combined with various quality enhancement methods.



**FIGURE A1** Coding efficiencies of two JointIQ-Net variations with different training methods. We jointly trained the image compression and enhancement subnetworks for one version and separately trained them for the other version.



**FIGURE A2** Coding efficiencies of the JointIQ-Net variations combined with various quality enhancement methods.

## APPENDIX B: ADDITIONAL IMPLEMENTATION DETAILS

The detailed structure of JointIQ-Net is presented in Figure 2, where  $N$  and  $M$  are set based on the corresponding  $\lambda$  values. The values of  $N$  and  $M$  for different  $\lambda$  values are listed in Table B1. We set  $G$ , which is the number of Gaussian PDFs for each representation, to three.

Therefore, the model parameter estimator  $f$  outputs  $9 \times M$  values for  $M$  representations of  $\hat{y}_i$  located at the specific spatial position of  $\hat{y}$ . To determine the global context, we set  $K$  to seven, which is approximately half of the maximum spatial distance between representations in  $\hat{y}$  when  $256 \times 256$  image patches are used for training. We utilized the global context only when the number of representations in the global context region was at least 30 to maintain the statistical significance of the global context. For fewer than 30 representations, we set all values in the global context to zero. For the GRDN in our final model, we set the number of GRDBs, RDBs in each GRDB, convolutional layers in each RDB, and kernels used by each convolutional layer to 4, 4, 8, and 64, respectively. Notably, the GRDN presented in Figure 8 is a lightweight version for which the above parameters were set to 4, 3, 3, and 32 for simulations with low complexity.

For training, we extracted 51,140  $256 \times 256$  patches in a non-overlapping manner from the CLIC [39] training image set. The batch size was set to eight and all models were trained using the ADAM optimizer [42] with its default settings. The models were trained using the individual initial learning rates listed in Table B1. We applied gradient decay by reducing the learning rate by half every 50,000 steps during the final 300,000 steps. To ensure suitable scaling of the  $\lambda$  range, Equation (B1) was used as an objective cost function in the actual implementation.

$$\mathcal{L} = \frac{\lambda}{W_y \cdot H_y \cdot 256} R + \frac{1-\lambda}{1000} D. \quad (\text{B1})$$

Here,  $W_y$  and  $H_y$  denote the weight and height of  $\mathbf{y}$ , respectively. To train the final models compared in Figure 6, which include the deeper  $Q$  (GRDN) subnetworks, we used the same  $256 \times 256$  input patches but randomly extracted patches  $96 \times 96$  in size from the outputs of the  $I$  subnetwork and then fed them into the  $Q$  subnetwork. The distortion term was calculated based on the corresponding area of the input patches and the rate term was calculated based on the corresponding  $6 \times 6$  area of

the  $16 \times 16$  spatial area of  $\hat{y}$ . To reduce training time, we utilized pre-trained  $I$  subnetworks for the models with lightweight  $Q$  subnetworks. First, we trained the deeper  $Q$  subnetwork using only the distortion term for 100 K iterations, and then further optimized the  $I$  and  $Q$  subnetworks in an end-to-end manner for an additional 1M iterations. When training the MS-SSIM-optimized version of our final model, we calculated MS-SSIM based on three different scales, rather than five widely used scales. However, it should be noted that we calculated all MS-SSIM values based on five scales for testing.

## APPENDIX C: EXPERIMENTAL RESULTS ON THE CLIC AND TECNICK IMAGE SETS

As described in the main paper, JointIQ-Net is the *first model* that surpasses the coding efficiency performance of the most recent and advanced image compression codec, namely, VVC intra-coding (VTM 7.1 [23]), which is nearly finalized for standardization. This appendix provides numerous experimental results that support the superiority of JointIQ-Net compared with SOTA image compression methods.

To assess the effectiveness of JointIQ-Net thoroughly, we performed experiments comparing our model and SOTA methods, namely, VVC intra-coding (VTM 7.1 [23]), BPG [19], and the model proposed by Lee and others [6], on two different image datasets, namely, the CLIC [39] validation set and the SAMPLING test set of the Tecnick [40] image set. Table B2 presents the BD-rate gains of the proposed JointIQ-Net versus VTM 7.1 [23] and BPG [19] on the CLIC [39] validation set and versus VTM 7.1 [23] and BPG [19] on the Tecnick [40] image set. It is clear that JointIQ-Net significantly outperformed the SOTA methods in terms of coding efficiency, outperforming VTM 7.1 [23] by averages of 4.85% and 7.12% on the CLIC [39] validation set and Tecnick [40] image set, respectively. It should also be noted that we utilized one line of padding for each input image (feature map) in a convolution layer when downscaling was needed.

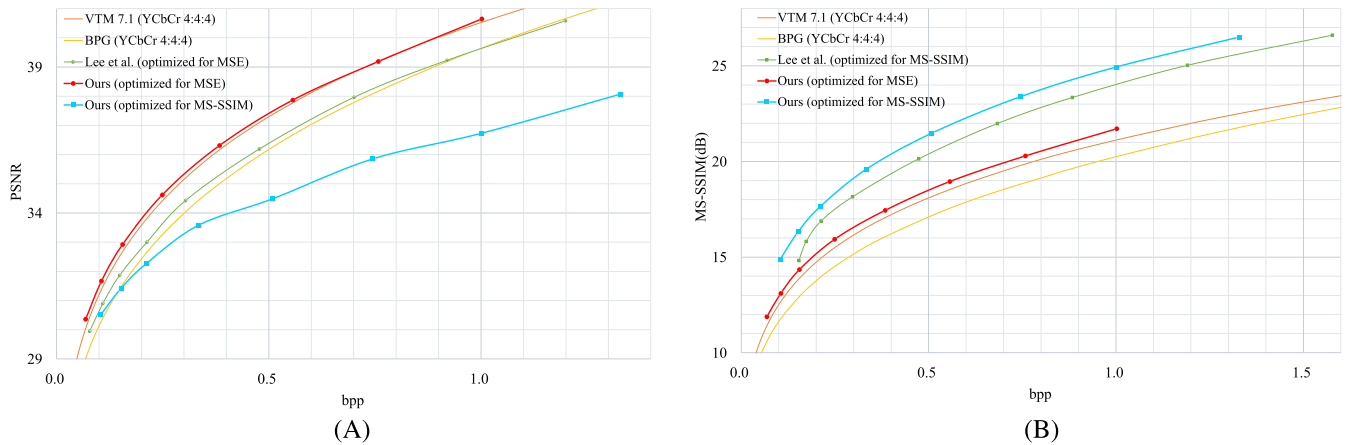
TABLE B1 Hyperparameter configurations for different  $\lambda$  values.

$\lambda$	$N$	$M$	No. of iterations	Initial learning rates
0.5	128	128	1.2M	1e-4
0.35	128	128	1.2M	1e-4
0.23	128	192	1.5M	1e-4
0.12	192	256	1.5M	1e-4
0.06	192	420	2.0M	5e-5
0.03	192	420	2.0M	5e-5
0.017	256	600	3.0M	5e-5
0.01	256	600	3.0M	3e-5

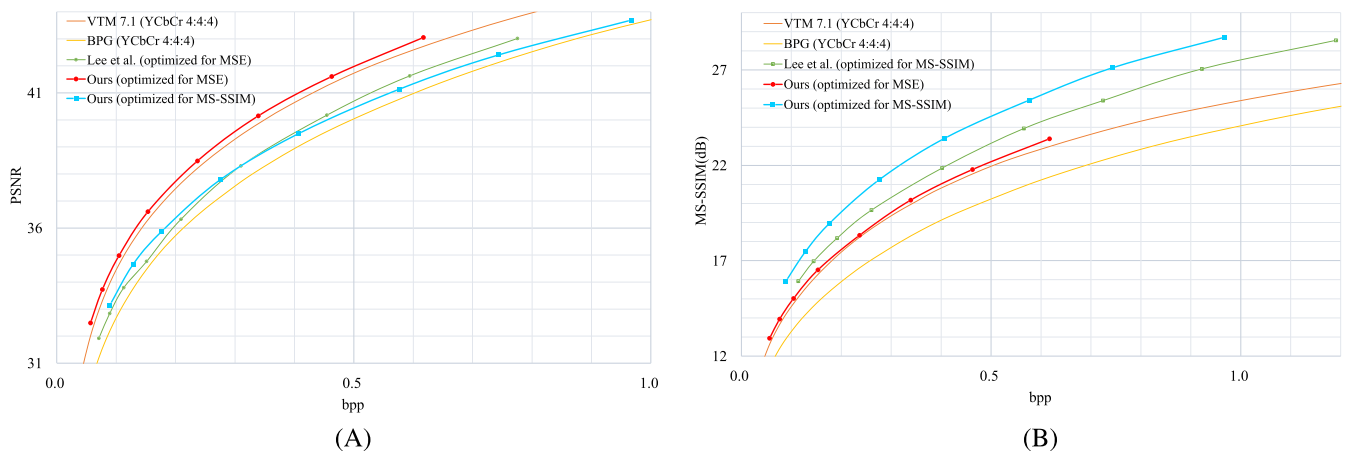
**TABLE B2** BD-rate gains of the proposed JointIQ-Net against VVC intra-coding (VTM 7.1 [23]) and BPG [19] on the CLIC [39] validation set and against VTM 7.1 [23] and BPG [19] on the Tecnick [40] image set.

	CLIC [39]			Tecnick [40]		
	VVC intra-coding [23]	BPG [19]	Lee [6]	VVC intra-coding [23]	BPG [19]	Lee [6]
MSE opt.	4.85%	28.16%	21.03%	7.12%	35.93%	28.21%
MS-SSIM opt.	52.60%	62.19%	21.25%	37.85%	54.78%	21.97%

Note: The third row presents the coding gains of our MSE-optimized model in terms of the PSNR-based BD rate, and the fourth row presents the coding gains of our MS-SSIM-optimized model in terms of the MS-SSIM-based BD rate.



**FIGURE B1** RD curves of the proposed JointIQ-Net and SOTA methods VTM 7.1 [23] and BPG [19] on the CLIC validation image dataset [39]. The left and right plots represent RD curves in terms of PSNR and MS-SSIM, respectively.



**FIGURE B2** RD curves of JointIQ-Net and the SOTA methods VTM 7.1 [23] and BPG [19] on the TECNICK image dataset [40]. The left and right plots represent RD curves in terms of PSNR and MS-SSIM, respectively.

Specifically, when the numbers of horizontal and vertical lines in an input image (feature map) were odd and the input image (feature map) needed to be downsampled, one additional horizontal line and one additional vertical line were added as padding to the bottommost and rightmost areas, respectively, to ensure even line numbers prior to horizontal and vertical downsampling. Furthermore, the sizes of the file headers, which indicate the original input sizes, were included in the bpp calculation. To perform evaluation using the CLIC [39] validation set and

SAMPLING test set of the Tecnick [40] image set, we limited the maximum horizontal and vertical distances, which  $c_i^m$  is calculated over, to 16.

Figures B1 and B2 present the coding efficiency curves corresponding to the results in Table B2 for the CLIC [39] validation set and SAMPLING test set of the Tecnick [40] image set, respectively. It is noteworthy that in Figures B1 and B2, one can see that JointIQ-Net outperformed all the SOTA methods over the entire bpp range.