Variable Rate Deep Image Compression With a Conditional Autoencoder

Yoojin Choi, Mostafa El-Khamy, Jungwon Lee
SoC R&D, Samsung Semiconductor Inc., San Diego, CA 92121, USA
{yoojin.c,mostafa.e,jungwon2.lee}@samsung.com

Abstract

In this paper, we propose a novel variable-rate learned image compression framework with a conditional autoencoder. Previous learning-based image compression methods mostly require training separate networks for different compression rates so they can yield compressed images of varying quality. In contrast, we train and deploy only one variable-rate image compression network implemented with a conditional autoencoder. We provide two rate control parameters, i.e., the Lagrange multiplier and the quantization bin size, which are given as conditioning variables to the network. Coarse rate adaptation to a target is performed by changing the Lagrange multiplier, while the rate can be further fine-tuned by adjusting the bin size used in quantizing the encoded representation. Our experimental results show that the proposed scheme provides a better rate-distortion trade-off than the traditional variable-rate image compression codecs such as JPEG2000 and BPG. Our model also shows comparable and sometimes better performance than the state-of-the-art learned image compression models that deploy multiple networks trained for varying rates.

1. Introduction

Image compression is an application of data compression for digital images to lower their storage and/or transmission requirements. Transform coding [8] has been successful to yield practical and efficient image compression algorithms such as JPEG [27] and JPEG2000 [18]. The transformation converts an input to a latent representation in the transform domain where lossy compression (that is typically a combination of quantization and lossless source coding) is more amenable and more efficient. For example, JPEG utilizes the discrete cosine transform (DCT) to convert an image into a sparse frequency domain representation. JPEG2000 replaces DCT with an enhanced discrete wavelet transform.

Deep learning is now leading many performance breakthroughs in various computer vision tasks [13]. Along with this revolutionary progress of deep learning, learned image compression also has derived significant interests [1, 3, 4, 9, 14–16, 19, 23, 24]. In particular, non-linear transform coding designed with deep neural networks has advanced to outperform the classical image compression codecs sophisticatedly designed and optimized by domain experts, e.g., BPG [5], which is a still image version of the high efficiency video codec (HEVC) standard [22]—we note that very recently, only a few of the learning-based image compression schemes have reached the performance of the state-of-the-art BPG codec on peak signal-to-noise ratio (PSNR), a metric based on mean squared error (MSE) [14, 16].

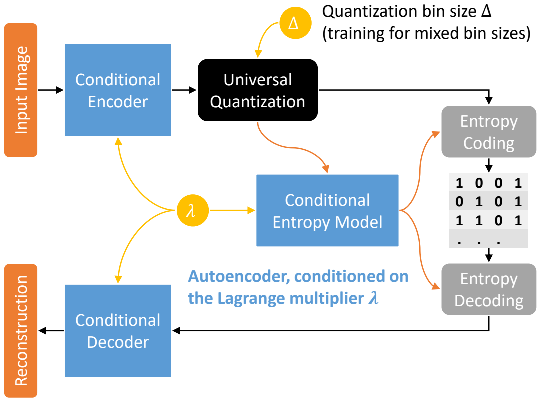
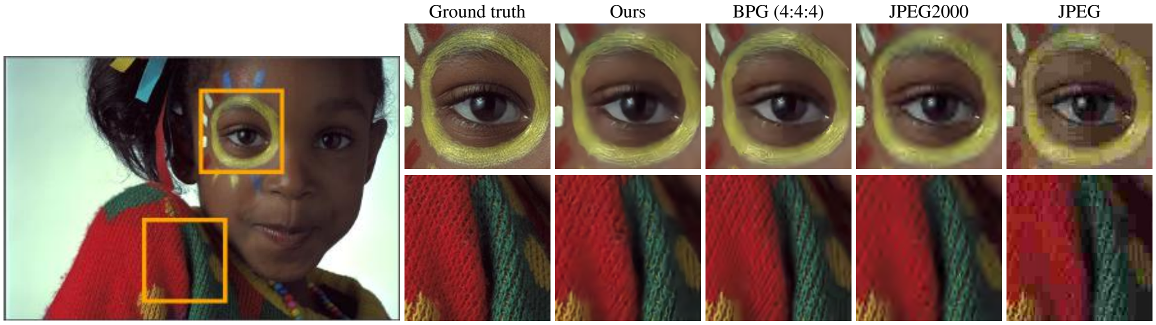


Figure 1: Our variable-rate image compression model. We provide two knobs to vary the rate. First, we employ a conditional autoencoder, conditioned on the Lagrange multiplier λ that adapts the rate, and optimize the rate-distortion Lagrangian for various λ values in one conditional model. Second, we train the model for mixed values of the quantization bin size Δ so we can vary the rate by changing Δ.

The resemblance of non-linear transform coding and autoencoders has been established and exploited for image compression in [3, 23]—an encoder transforms an image (a set of pixels) into a latent representation in a lower dimensional space, and a decoder performs an approximate inverse transform that converts the latent representation back to the image. The transformation is desired to yield a latent representation with the smallest entropy, given a distortion level, since the entropy is the minimum rate achievable with lossless entropy source coding [7, Section 5.3]. In practice,



	Ours	BPG (4:4:4)	JPEG2000	JPEG
Bits per pixel (BPP)	0.1697	0.1697	0.1702	0.1775
PSNR (dB)	32.2332	31.9404	30.3140	27.3389
MS-SSIM	0.9602	0.9539	0.9369	0.8669

Figure 2: PSNR and MS-SSIM comparison of our model and classical image compression algorithms (BPG, JPEG2000, and JPEG). We adapt the rate by changing the Lagrange multiplier λ and the quantization bin size Δ to match the rate of BPG. In this example, we observe 0.3 dB PSNR gain over the state-of-the-art BPG codec. A perceptual measure, MS-SSIM, also improves. Visually, our method provides better quality with less artifacts than the classical image compression codecs.

however, it is generally not straightforward to calculate and optimize the exact entropy of a latent representation. Hence, the rate-distortion (R-D) trade-off is optimized by minimizing an entropy estimate of a latent representation provided by an autoencoder at a target quality. To improve compression efficiency, recent methods have focused on developing accurate entropy estimation models [1, 4, 14–16] with sophisticated density estimation techniques such as variational Bayes and autoregressive context modeling.

Given a model that provides an accurate entropy estimate of a latent representation, the previous autoencoder-based image compression frameworks optimize their networks by minimizing the weighted sum of the R-D pairs using the method of Lagrange multipliers. The Lagrange multiplier λ introduced in the Lagrangian (see (2)) is treated as a hyper-parameter to train a network for a desired trade-off between the rate and the quality of compressed images. This implies that one needs to train and deploy separate networks for rate adaptation. One way is to re-train a network while varying the Lagrange multiplier. However, this is impractical when we operate at a broad range of the R-D curve with fine resolution and the size of each network is large.

In this paper, we suggest training and deploying only one variable-rate image compression network that is capable of rate adaptation. In particular, we propose a conditional autoencoder, conditioned on the Lagrange multiplier, i.e., the network takes the Lagrange multiplier as an input and produces a latent representation whose rate depends on the input value. Moreover, we propose training the network with mixed quantization bin sizes, which allows us to adapt the rate by adjusting the bin size applied to the quantization of a latent representation. Coarse rate adaptation to a target is achieved by varying the Lagrange multiplier in the conditional model, while fine rate adaptation is done by tuning the quantization bin size. We illustrate our variable-rate image compression model in Figure 1.

Conditional autoencoders have been used for conditional generation [21, 26], where their conditioning variables are typically labels, attributes, or partial observations of the target output. However, our conditional autoencoder takes a hyper-parameter, i.e., the Lagrange multiplier, of the optimization problem as its conditioning variable. We basically solve multiple objectives using one conditional network, instead of solving them individually using separate non-conditional networks (each optimized for one objective), which is new to the best of our knowledge.

We also note that variable-rate models using recurrent neural networks (RNNs) were proposed in [9, 24]. However, the RNN-based models require progressive encoding and decoding, depending on the target image quality. The increasing number of iterations to obtain a higher-quality image is not desirable in certain applications and platforms. Our variable-rate model is different from the RNN-based models. Our model is based on a conditional autoencoder that needs no multiple iterations, while the quality is controlled by its conditioning variables, i.e., the Lagrange multiplier and the quantization bin size. Our method also shows superior performance over the RNN-based models in [9,24].

We evaluate the performance of our variable-rate image compression model on the Kodak image dataset [12] for both the objective image quality metric, PSNR, and a perceptual score measured by the multi-scale structural similarity (MS-SSIM) [28]. The experimental results show that our variable-rate model outperforms BPG in both PSNR and MS-SSIM metrics; an example from the Kodak dataset is shown in Figure 2. Moreover, our model shows a comparable and sometime better R-D trade-off than the state-of-the-art learned image compression models [14, 16] that outperform BPG by deploying multiple networks trained for different target rates.

2. Preliminary

We consider a typical autoencoder architecture consisting of encoder $f_\phi(\mathbf{x})$ and decoder $g_\theta(\mathbf{z})$, where $\mathbf{x}$ is an input image and $\mathbf{z} = \mathrm{round}_\Delta(f_\phi(\mathbf{x}))$ is a quantized latent representation encoded from the input $\mathbf{x}$ with quantization bin size Δ; we let $\mathrm{round}_\Delta(x) = \Delta\,\mathrm{round}(x/\Delta)$, where round denotes element-wise rounding to the nearest integer. For now, we fix $\Delta = 1$. Lossless entropy source coding, e.g., arithmetic coding [7, Section 13.3], is used to generate a compressed bitstream from the quantized representation $\mathbf{z}$. Let $\mathbb{E}_{p(x)}[A(x)] = \int A(x)p(x)dx$, where $p(x)$ is the probability density function of x.

Deterministic quantization. Suppose that we take entropy source coding for the quantized latent variable $\mathbf{z}$ and achieve its entropy rate. The rate R and the squared L2 distortion D (i.e., the MSE loss) are given by

$$\begin{aligned} R_\phi &= \sum_{\mathbf{z}} -P_\phi(\mathbf{z}) \log_2 P_\phi(\mathbf{z}), \\ D_{\phi,\theta} &= \mathbb{E}_{p(\mathbf{x})}[\|\mathbf{x} - g_\theta(\mathrm{round}_\Delta(f_\phi(\mathbf{x})))\|_2^2], \end{aligned} \tag{1}$$

where $p(\mathbf{x})$ is the probability density function of all natural images, and $P_\phi(\mathbf{z})$ is the probability mass function of $\mathbf{z}$ induced from encoder $f_\phi(\mathbf{x})$ and round_Δ, which satisfies $P_\phi(\mathbf{z}) = \int p(\mathbf{x})\delta(\mathbf{z} - \mathrm{round}_\Delta(f_\phi(\mathbf{x})))d\mathbf{x}$, where δ denotes the Dirac delta function. Using the method of Lagrange multipliers, the R-D optimization problem is given by

$$\min_{\phi,\theta} \left\{ D_{\phi,\theta} + \lambda R_\phi \right\}, \tag{2}$$

for $\lambda > 0$; the scalar factor λ in the Lagrangian is called a Lagrange multiplier. The Lagrange multiplier is the factor that selects a specific R-D trade-off point (e.g., see [17]).

Relaxation with universal quantization. The rate and the distortion provided in (1) are not differentiable for network parameter ϕ, due to $P_\phi(\mathbf{z})$ and round_Δ, and thus it is not straightforward to optimize (2) through gradient descent. It was proposed in [3] to model the quantization error as additive uniform stochastic noise to relax the optimization of (2). The same technique was adopted in [4, 14, 16]. In this paper, we instead propose employing universal quantization [29, 30] to relax the problem (see Remark 2).

Universal quantization dithers every element of $f_\phi(\mathbf{x})$ with one common uniform random variable as follows:

$$\mathbf{z} = \mathrm{round}_\Delta(f_\phi(\mathbf{x}) + \mathbf{u}) - \mathbf{u}, \quad \mathbf{u} = [U, U, \ldots, U], \tag{3}$$

where the dithering vector $\mathbf{u}$ consists of repetitions of a single uniform random variable U with support $[-\Delta/2, \Delta/2]$. We fix $\Delta = 1$ just for now. In each dimension, universal quantization is effectively identical in distribution to adding uniform noise independent of the source, although the noise induced from universal quantization is dependent across dimensions. Note that universal quantization is approximated as a linear function of the unit slope (of gradient 1) in the backpropagation of the network training.

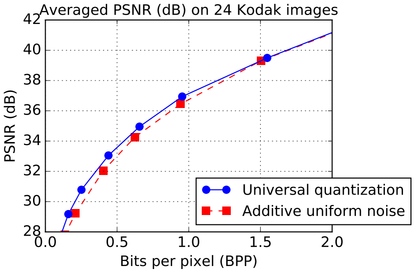


Figure 3: The network trained with universal quantization gives higher PSNR than the one trained with additive uniform noise in our experiments on 24 Kodak images.

Remark 1. To our knowledge, we are the first to adopt universal quantization in the framework of training image compression networks. In [6], universal quantization was used for efficient weight compression of deep neural networks, which is different from our usage here. We observed from our experiments that our relaxation with universal quantization provides some gain over the conventional method of adding independent uniform noise (see Figure 3).

Differentiable R-D cost function. Under the relaxation with universal quantization, similar to (1), the rate and the distortion can be expressed as below:

$$\begin{aligned} R_\phi &= \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x})}[-\log_2 p_\phi(\mathbf{z})], \\ D_{\phi,\theta} &= \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x})}[\|\mathbf{x} - g_\theta(\mathbf{z})\|_2^2], \end{aligned} \tag{4}$$

where $p_\phi(\mathbf{z}) = \int p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x})d\mathbf{x}$. The stochastic quantization model makes $\mathbf{z}$ have a continuous density $p_\phi(\mathbf{z})$, which is a continuous relaxation of $P_\phi(\mathbf{z})$, but still $p_\phi(\mathbf{z})$ is usually intractable to compute. Thus, we further adopt approximation of $p_\phi(\mathbf{z})$ to a tractable density $q_\theta(\mathbf{z})$ that is differentiable with respect to $\mathbf{z}$ and θ. Then, it follows that

$$\begin{aligned} R_\phi &= \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x})}[-\log_2 q_\theta(\mathbf{z})] - \mathrm{KL}(p_\phi(\mathbf{z})||q_\theta(\mathbf{z})) \\ &\leq \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x})}[-\log_2 q_\theta(\mathbf{z})] \triangleq R_{\phi,\theta}, \end{aligned} \tag{5}$$

where KL denotes Kullback-Leibler (KL) divergence (e.g., see [7, p. 19]); the equality in $\leq$ holds when $p_\phi(\mathbf{z}) = q_\theta(\mathbf{z})$. The choice of $q_\theta(\mathbf{z})$ in our implementation is deferred to Section 4 (see (12)–(14)).

From (2) and (4), approximating R_ϕ by its upperbound $R_{\phi,\theta}$ in (5), the R-D optimization problem reduces to

$$\min_{\phi,\theta} \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x})}[\|\mathbf{x} - g_\theta(\mathbf{z})\|_2^2 - \lambda \log_2 q_\theta(\mathbf{z})], \tag{6}$$

for $\lambda > 0$. Optimizing a network for different values of λ, one can trade off the quality against the rate.

Remark 2. The objective function in (6) has the same form as auto-encoding variational Bayes [11], given that the posterior $p_\phi(\mathbf{z}|\mathbf{x})$ is uniform. This relation was already established in the previous works, and detailed discussions can be

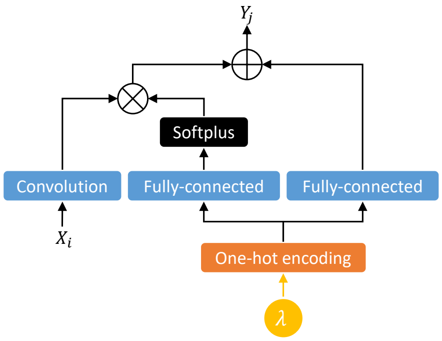


Figure 4: Conditional convolution, conditioned on the Lagrange multiplier λ, which produces a different output depending on the input Lagrange multiplier λ.

found in [3,4]. Our contribution in this section is to deploy universal quantization (see (3)) to guarantee that the quantization error is uniform and independent of the source distribution, instead of artificially adding uniform noise, when generating random samples of $\mathbf{z}$ from $p_\phi(\mathbf{z}|\mathbf{x})$ in Monte Carlo estimation of (6).

3. Variable rate image compression

To adapt the quality and the rate of compressed images, we basically need to optimize the R-D Lagrange function in (6) for varying values of the Lagrange multiplier λ. That is, one has to train multiple networks or re-train a network while varying the Lagrange multiplier λ. Training and deploying multiple networks are not practical, in particular when we want to cover a broad range of the R-D curve with fine resolution, and each network is of a large size. In this section, we develop a variable-rate model that can be deployed once and can be used to produce compressed images of varying quality with different rates, depending on user's requirements, with no need of re-training.

3.1. Conditional autoencoder

To avoid training and deploying multiple networks, we propose training one conditional autoencoder, conditioned on the Lagrange multiplier λ. The network takes λ as a conditioning input parameter, along with the input image, and produces a compressed image with varying rate and distortion depending on the conditioning value of λ. To this end, the rate and distortion terms in (4) and (5) are altered into

$$R_{\phi,\theta}(\lambda) = \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x},\lambda)}[-\log_2 q_\theta(\mathbf{z}|\lambda)],$$
$$D_{\phi,\theta}(\lambda) = \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x},\lambda)}[\|\mathbf{x} - g_\theta(\mathbf{z},\lambda)\|_2^2],$$

for $\lambda \in \Lambda$, where Λ is a pre-defined finite set of Lagrange multiplier values, and then we minimize the following combined objective function:

$$\min_{\phi,\theta} \sum_{\lambda\in\Lambda} \left(D_{\phi,\theta}(\lambda) + \lambda R_{\phi,\theta}(\lambda)\right). \tag{7}$$

To implement a conditional autoencoder, we develop the conditional convolution, conditioned on the Lagrange multiplier λ, as shown in Figure 4. Let X_i be a 2-dimensional (2-D) input feature map of channel i and Y_j be a 2-D output feature map of channel j. Let $W_{i,j}$ be a 2-D convolutional kernel for input channel i and output channel j. Our conditional convolution yields

$$Y_j = s_j(\lambda) \sum_i X_i * W_{i,j} + b_j(\lambda), \tag{8}$$

where $*$ denotes 2-D convolution. The channel-wise scaling factor and the additive bias term depend on λ by

$$\begin{aligned} s_j(\lambda) &= \mathrm{softplus}(u_j^T \,\mathrm{onehot}_\Lambda(\lambda)), \\ b_j(\lambda) &= v_j^T \,\mathrm{onehot}_\Lambda(\lambda), \end{aligned} \tag{9}$$

where u_j and v_j are the fully-connected layer weight vectors of length $|\Lambda|$ for output channel j; T denotes the transpose, $\mathrm{softplus}(x) = \log(1 + e^x)$, and $\mathrm{onehot}_\Lambda(\lambda)$ is one-hot encoding of λ over Λ.

Remark 3. The proposed conditional convolution is similar to the one proposed by conditional PixelCNN [26]. In [26], conditioning variables are typically labels, attributes, or partial observations of the target output, while our conditioning variable is the Lagrange multiplier, which is the hyperparameter that trades off the quality against the rate in the compression problem. A gated-convolution structure is presented in [26], but we develop a simpler structure so that the additional computational cost of conditioning is marginal.

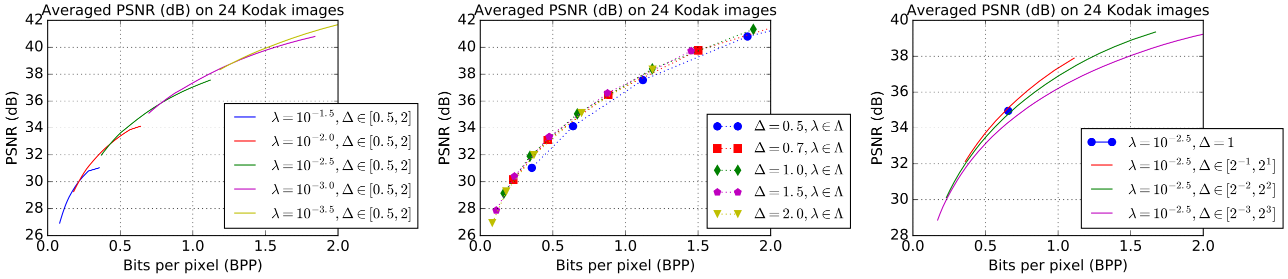
3.2. Training with mixed bin sizes

We established a variable-rate conditional autoencoder model, conditioned on the Lagrange multiplier λ in the previous subsection, but only finite discrete points in the R-D curve can be obtained from it, since λ is selected from a pre-determined finite set Λ.[1] To extend the coverage to the whole continuous range of the R-D curve, we develop another (continuous) knob to control the rate, i.e., the quantization bin size.

Recall that in the previous R-D formulation (1), we fixed the quantization bin size $\Delta = 1$, i.e., we simply used round for quantization. In actual inference, we can change the bin size to adapt the rate—the larger the bin size, the lower the rate. However, the performance naturally suffers from mismatched bin sizes in training and inference. For a trained network to be robust and accurate for varying bin sizes, we propose training (or fine-tuning) it with mixed bin sizes.

In training, we draw a uniform noise in (3) for various noise levels, i.e., for random Δ. The range of Δ and the mixing distribution within the range are design choices. In our experiments, we choose $\Delta = 2^b$, where b is uniformly

[1] The conditioning part can be modified to take continuous λ values, which however did not produce good results in our trials.



(a) Vary $\Delta \in [0.5, 2]$ for fixed $\lambda \in \Lambda$

(b) Vary $\lambda \in \Lambda$ for fixed Δ

(c) Vary the mixing range of Δ in training

Figure 5: In (a,b), we show how we can adapt the rate in our variable-rate model by changing the Lagrange multiplier λ and the quantization bin size Δ. In (a), we vary Δ within $[0.5, 2]$ for each fixed $\lambda \in \Lambda$ in (15). In (b), we change λ in Λ while fixing Δ for some selected values. In (c), we compare PSNR when models are trained for mixed bin sizes of different ranges.

drawn from $[-1, 1]$ so we can cover $\Delta \in [0.5, 2]$. The larger the range of b, we optimize a network for a broader range of the R-D curve, but the performance also degrades. In Figure 5(c), we compare the R-D curves obtained from the networks trained with mixed bin sizes of different ranges; we used fixed $\lambda = 10^{-2.5}$ in training the networks just for this experiment. We found that mixing bin sizes in $\Delta \in [0.5, 2]$ yields the best performance, although the coverage is limited, which is not a problem since we can cover large-scale rate adaptation by changing the input Lagrange multiplier in our conditional model (see Figure 5 (a,b)).

In summary, we solve the following optimization:

$$\min_{\phi,\theta} \sum_{\lambda \in \Lambda} \mathbb{E}_{p(\Delta)}[D_{\phi,\theta}(\lambda, \Delta) + \lambda R_{\phi,\theta}(\lambda, \Delta)], \tag{10}$$

where $p(\Delta)$ is a pre-defined mixing density for Δ, and

$$\begin{aligned} R_{\phi,\theta}(\lambda, \Delta) &= \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x},\lambda,\Delta)}[-\log_2 q_\theta(\mathbf{z}|\lambda, \Delta)], \\ D_{\phi,\theta}(\lambda, \Delta) &= \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x},\lambda,\Delta)}[\|\mathbf{x} - g_\theta(\mathbf{z}, \lambda)\|_2^2]. \end{aligned} \tag{11}$$

Remark 4. In training, we compute neither the summation over $\lambda \in \Lambda$ nor the expectation over $p(\Delta)$ in (10). Instead, we randomly select λ uniformly from Λ and draw Δ from $p(\Delta)$ for each image to compute its individual R-D cost, and then we use the average R-D cost per batch as the loss for gradient descent, which makes the training scalable.

3.3. Inference

Rate adaptation. The rate increases, as we decrease the Lagrange multiplier λ and/or the quantization bin size Δ. In Figure 5(a,b), we show how the rate varies as we change λ and Δ. In (a), we change Δ within $[0.5, 2]$ for each fixed $\lambda \in \Lambda$ from (15). In (b), we vary λ in Λ while fixing Δ for some selected values. Given a user's target rate, large-scale discrete rate adaptation is achieved by changing λ, while fine continuous rate adaptation can be performed by adjusting Δ for fixed λ. When the R-D curves overlap at the target rate (e.g., see 0.5 BPP in Figure 5(a)), we select the combination of λ and Δ that produces better performance.[2]

Compression. After selecting $\lambda \in \Lambda$, we do one-hot encoding of λ and use it in all conditional convolutional layers to encode a latent representation of the input. Then, we perform regular deterministic quantization on the encoded representation with the selected quantization bin size Δ. The quantized latent representation is then finally encoded into a compressed bitstream with entropy coding, e.g., arithmetic coding; we additionally need to store the values of the conditioning variables, λ and Δ, used in encoding.

Decompression. We decode the compressed bitstream. We also retrieve λ and Δ used in encoding from the compressed bitstream. We restore the quantized latent representation from the decoded integer values by multiplying them with the quantization bin size Δ. The restored latent representation is then fed to the decoder to reconstruct the image. The value of λ used in encoding is again used in all deconvolutional layers, for conditional generation.

4. Refined probabilistic model

In this section, we discuss how we refine the baseline model in the previous section to improve the performance. The model refinement is orthogonal to the rate adaptation schemes in Section 3. From (11), we introduce a secondary latent variable $\mathbf{w}$ that depends on $\mathbf{x}$ and $\mathbf{z}$ to yield

$$\begin{aligned} R_{\phi,\theta}(\lambda, \Delta) &= \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x},\lambda,\Delta)p_\phi(\mathbf{w}|\mathbf{z},\mathbf{x},\lambda,\Delta)} \\ &\qquad [-\log_2(q_\theta(\mathbf{w}|\lambda, \Delta)q_\theta(\mathbf{z}|\mathbf{w}, \lambda, \Delta))], \\ D_{\phi,\theta}(\lambda, \Delta) &= \mathbb{E}_{p(\mathbf{x})p_\phi(\mathbf{z}|\mathbf{x},\lambda,\Delta)p_\phi(\mathbf{w}|\mathbf{z},\mathbf{x},\lambda,\Delta)} \\ &\qquad [\|\mathbf{x} - g_\theta(\mathbf{z}, \mathbf{w}, \lambda)\|_2^2]. \end{aligned}$$

For compression, we encode $\mathbf{z}$ from $\mathbf{x}$, and then we further encode $\mathbf{w}$ from $\mathbf{z}, \mathbf{x}$. The encoded representations $\mathbf{z}, \mathbf{w}$ are entropy-coded based on $q_\theta(\mathbf{w}|\lambda, \Delta)$, $q_\theta(\mathbf{z}|\mathbf{w}, \lambda, \Delta)$, respectively. For decompression, given $q_\theta(\mathbf{w}|\lambda, \Delta)$, we decode $\mathbf{w}$, which is then used to compute $q_\theta(\mathbf{z}|\mathbf{w}, \lambda, \Delta)$ and to decode

[2] In practice, one can make a set of pre-selected combinations of λ and Δ, similar to the set of quality factors in JPEG or BPG.

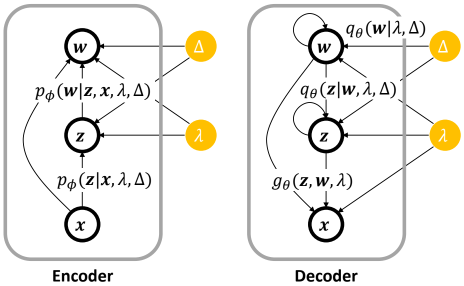

$q_\theta(\boldsymbol{w}|\lambda,\Delta)$
$\boldsymbol{w}$
Δ
$p_\phi(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{x},\lambda,\Delta)$
$q_\theta(\boldsymbol{z}|\boldsymbol{w},\lambda,\Delta)$
$\boldsymbol{z}$
λ
$p_\phi(\boldsymbol{z}|\boldsymbol{x},\lambda,\Delta)$
$g_\theta(\boldsymbol{z},\boldsymbol{w},\lambda)$
$\boldsymbol{x}$
Encoder
Decoder

Figure 6: A graph representation of our refined variable-rate image compression model.

$\mathbf{z}$. This model is further refined by introducing autoregressive models for $q_\theta(\mathbf{w}|\lambda,\Delta)$ and $q_\theta(\mathbf{z}|\mathbf{w},\lambda,\Delta)$ as below:

$$\begin{aligned} q_\theta(\mathbf{w}|\lambda,\Delta) &= \prod\nolimits_i q_\theta(w_i|w_{<i},\lambda,\Delta), \\ q_\theta(\mathbf{z}|\mathbf{w},\lambda,\Delta) &= \prod\nolimits_i q_\theta(z_i|z_{<i},\mathbf{w},\lambda,\Delta), \end{aligned} \tag{12}$$

where a_i is the i-th element of $\mathbf{a}$, and $a_{<i} = [a_1,\ldots,a_{i-1}]$. In Figure 6, we illustrate a graph representation of our refined variable-rate image compression model.

In our experiments, we use

$$q_\theta(z_i|z_{<i},\mathbf{w},\lambda,\Delta) = \int_{z_i-\Delta/2}^{z_i+\Delta/2} \frac{1}{\Delta\sigma_i} f_{\mathcal{N}}\left(\frac{x-\mu_i}{\sigma_i}\right) dx, \tag{13}$$

where $\mu_i = \mu_\theta(z_{<i},\mathbf{w},\lambda)$, $\sigma_i^2 = \sigma_\theta^2(z_{<i},\mathbf{w},\lambda)$, and $f_{\mathcal{N}}$ denotes the standard normal density; μ_θ and σ_θ^2 are parameterized with autoregressive neural networks, e.g., consisting of masked convolutions [26], which are also conditioned on λ as in Figure 4. Similarly, we let

$$q_\theta(w_i|w_{<i},\lambda,\Delta) = \int_{w_i-\Delta/2}^{w_i+\Delta/2} \frac{1}{\Delta\zeta_i} f_\psi\left(\frac{x-\nu_i}{\zeta_i}\right) dx, \tag{14}$$

where $\nu_i = \nu_\theta(w_{<i},\lambda)$, $\zeta_i^2 = \zeta_\theta^2(w_{<i},\lambda)$, and f_ψ is designed as a univariate density model parameterized with a neural network as described in [4, Appendix 6.1].

Remark 5. Setting aside the conditioning parts, the refined model can be viewed as a hierarchical autoencoder (e.g., see [25]). It is also similar to the one in [16] with the differences summarized in Appendix A.

5. Experiments

We illustrate the network architecture that we used in our experiments in Figure 7. We emphasize that all convolution (including masked convolution) blocks employ conditional convolutions (see Figure 4 in Section 3.1).

Training. For a training dataset, we used the ImageNet ILSVRC 2012 dataset [20]. We resized the training images so that the shorter of the width and the height is 256, and we extracted 256×256 patches at random locations. In addition to the ImageNet dataset, we used the training dataset provided in the Workshop and Challenge on Learned Image Compression (CLIC)[3]. For the CLIC training dataset, we extracted 256×256 patches at random locations without resizing. We used Adam optimizer [10] and trained a model for 50 epochs, where each epoch consists of 40k batches and the batch size is set to 8. The learning rate was set to be 10^{-4} initially, and we decreased the learning rate to 10^{-5} and 10^{-6} at 20 and 40 epochs, respectively.

We pre-trained a conditional model that can be conditioned on 5 different values of the Lagrange multiplier in Λ for fixed bin size $\Delta = 1$, where

$$\Lambda = \{10^{-1.5}, 10^{-2.0}, 10^{-2.5}, 10^{-3.0}, 10^{-3.5}\}. \tag{15}$$

In pre-training, we used the MSE loss. Then, we re-trained the model for mixed bin sizes; the quantization bin size Δ is selected randomly from $\Delta = 2^b$, where b is drawn uniformly between -1 and 1 so that we cover $\Delta \in [0.5, 2]$. In the re-training with mixed bin sizes, we used one of MSE, MS-SSIM and combined MSE+MS-SSIM losses (see Figure 9). We used the same training datasets and the same training procedure for pre-training and re-training. We also trained multiple fixed-rate models for fixed $\lambda \in \Lambda$ and fixed $\Delta = 1$ for comparison.

Experimental results. We compare the performance of our variable-rate model to the state-of-the-art learned image compression models from [4, 9, 14–16, 19] and the classical state-of-the-art variable-rate image compression codec, BPG [5], on the Kodak image set [12]. Some of the previous models were optimized for MSE, and some of them were optimized for a perceptual measure, MS-SSIM. Thus, we compare both measures separately in Figure 8. In particular, we included the results for the RNN-based variable-rate compression model in [9], which were obtained from [4]. All the previous works in Figure 8, except [9], trained multiple networks to get the multiple points in their R-D curves.

For our variable-rate model, we plotted 5 curves of the same blue color for PSNR and MS-SSIM, respectively, in Figure 8. Each curve corresponds to one of 5 Lagrange multiplier values in (15). For each $\lambda \in \Lambda$, we varied the quantization bin size Δ in $[0.5, 2]$ to get each curve. Our variable-rate model outperforms BPG in both PSNR and MS-SSIM measures. It also performs comparable overall and better in some cases than the state-of-the-art learned image compression models [14, 16] that outperform BPG by deploying multiple networks trained for varying rates.

Our model shows superior performance over the RNN-based variable-rate model in [9]. The RNN-based model requires multiple encoding/decoding iterations at high rates, implying the complexity increases as more iterations are needed to achieve better quality. In contrast, our model uses single iteration, i.e., the encoding/decoding complexity is fixed, for any rates. Moreover, our model can produce

[3] https://www.compression.cc

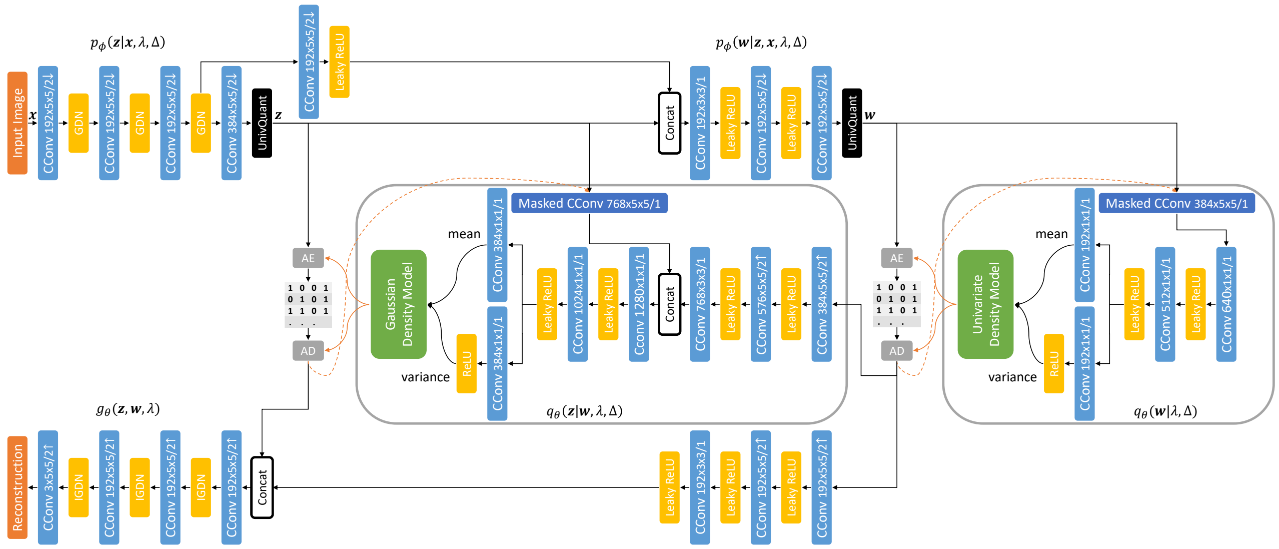


Figure 7: **UnivQuant** denotes universal quantization with the quantization bin size Δ. **AE** and **AD** are arithmetic encoding and decoding, respectively. **Concat** implies concatenation. **GDN** stands for generalized divisive normalization, and **IGDN** is inverse GDN [2]. The convolution parameters are denoted as # filters × kernel height × kernel width / stride, where ↑ and ↓ indicate upsampling and downsampling, respectively. **CConv** denotes conditional convolution, conditioned on the Lagrange multiplier λ (see Figure 4). All convolution and masked convolution blocks employ conditional convolutions. Upsampling convolutions are implemented as the deconvolution. Masked convolutions are implemented as in [26].

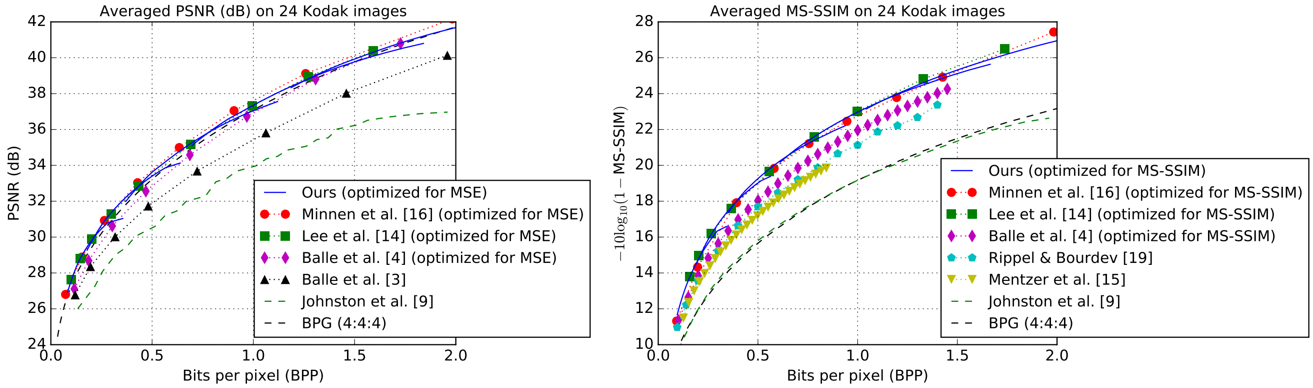


Figure 8: PSNR and MS-SSIM comparison to the state-of-the-art image compression models on 24 Kodak images. As in Figure 5(a), we plotted 5 curves from our variable-rate model for 5 Lagrange multiplier values in Λ of (15) and $\Delta \in [0.5, 2]$.

any point in the R-D curve with infinitely fine resolution by tuning the continuous rate-adaptive parameter, the quantization bin size Δ. However, the RNN-based model can produce only finite points in the R-D curve, depending on how many bits it encodes in each recurrent stage.

In Figure 9, we compare our variable-rate networks optimized for MSE, MS-SSIM and combined MSE+MS-SSIM losses, respectively. We also plotted the results from our fixed-rate networks trained for fixed λ and Δ. Observe that our variable-rate network performs very near to the ones individually optimized for fixed λ and Δ. Here, we emphasize that our variable-rate network optimized for MSE performs better than BPG in both PNSR and MS-SSIM measures.

Figure 10 shows the compressed images generated from our variable-rate model to assess their visual quality. We also depicted the number of bits (implicitly) used to represent each element of $\mathbf{z}$ and $\mathbf{w}$ in arithmetic coding, which are $-\log_2(\Delta q_\theta(z_i|z_{<i},\mathbf{w}))$ and $-\log_2(\Delta q_\theta(w_i|w_{<i}))$, respectively, in (12)–(14). We randomly selected two and four channels from $\mathbf{z}$ and $\mathbf{w}$, respectively, and showed the code length for each latent representation value in the figure. As we change conditioning parameters λ and Δ, we can adapt the arithmetic code length that determines the rate of the latent representation. Observe that the smaller the values of λ and/or Δ, the resulting latent representation requires more bits in arithmetic coding and the rate increases, as expected.

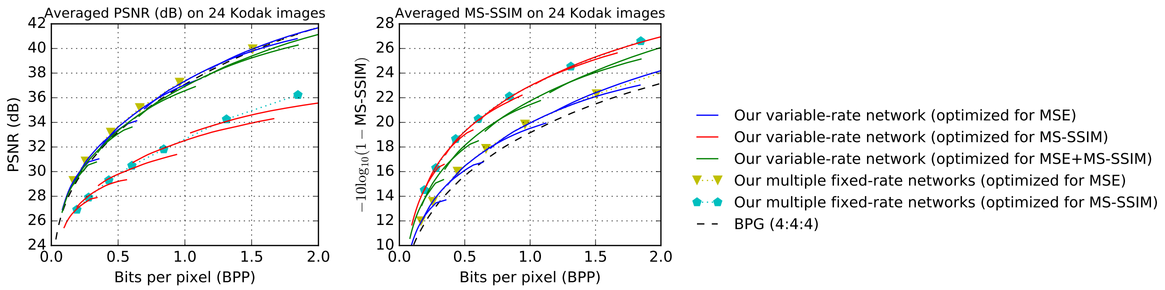
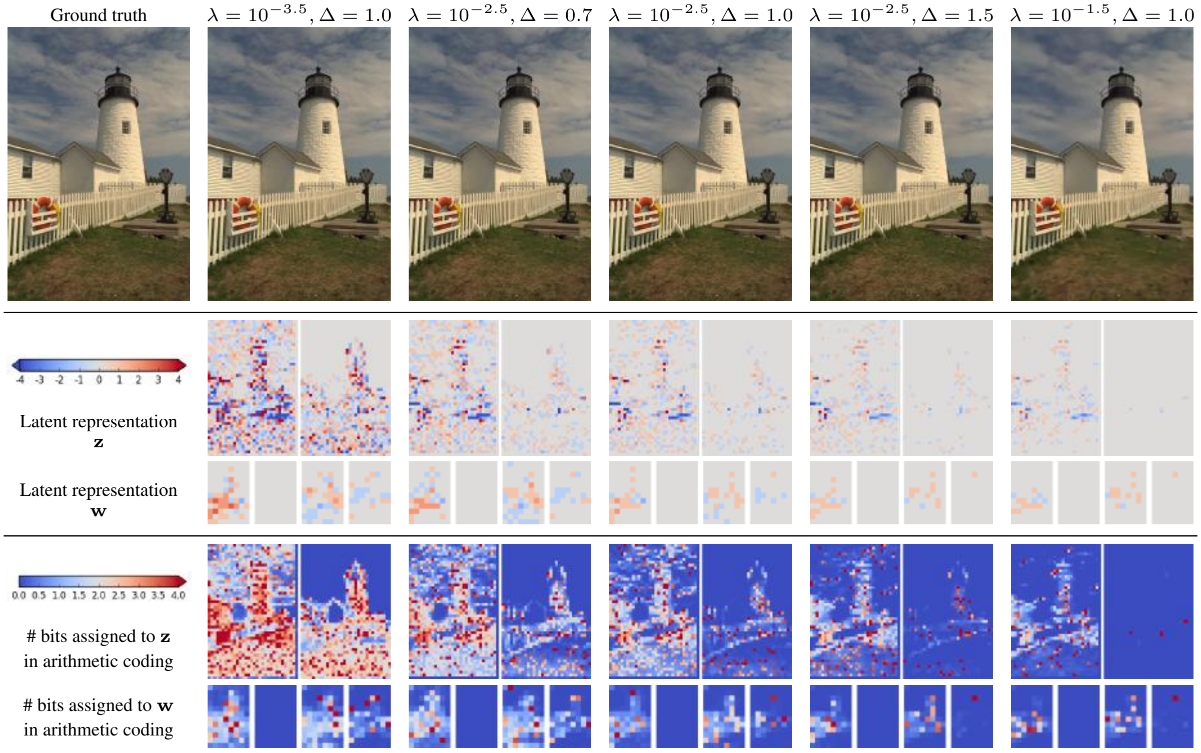


Figure 9: PSNR and MS-SSIM comparison on 24 Kodak images for our variable-rate and fixed-rate networks when they are optimized for MSE, MS-SSIM and combined MSE+MS-SSIM losses, respectively. In particular, we note that our variable-rate network optimized for MSE outperforms BPG in both PNSR and MS-SSIM measures.


Ground truth
$\lambda = 10^{-3.5}, \Delta = 1.0$
$\lambda = 10^{-2.5}, \Delta = 0.7$
$\lambda = 10^{-2.5}, \Delta = 1.0$
$\lambda = 10^{-2.5}, \Delta = 1.5$
$\lambda = 10^{-1.5}, \Delta = 1.0$
-4 -3 -2 -1 0 1 2 3 4
Latent representation $\mathbf{z}$
Latent representation $\mathbf{w}$
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
# bits assigned to $\mathbf{z}$ in arithmetic coding
# bits assigned to $\mathbf{w}$ in arithmetic coding

Bits per pixel (BPP) PSNR (dB) MS-SSIM	1.8027 41.3656 0.9951	0.8086 36.2535 0.9863	0.6006 34.8283 0.9819	0.4132 33.1478 0.9737	0.1326 29.2833 0.9249

Figure 10: Our variable-rate image compression outputs for different values of λ and Δ. We also depicted the value and the number of bits assigned to each element of latent representations $\mathbf{z}$ and $\mathbf{w}$ in arithmetic coding, respectively.

6. Conclusion

This paper proposed a variable-rate image compression framework with a conditional autoencoder. Unlike the previous learned image compression methods that train multiple networks to cover various rates, we train and deploy one variable-rate model that provides two knobs to control the rate, i.e., the Lagrange multiplier and the quantization bin size, which are given as input to the conditional autoencoder model. Our experimental results showed that the proposed scheme provides better performance than the classical image compression codecs such as JPEG2000 and BPG. Our method also showed comparable and sometimes better performance than the recent learned image compression methods that outperform BPG but need multiple networks trained for different compression rates. We finally note that the proposed conditional neural network can be adopted in deep learning not only for image compression but also in general to solve any optimization problem that can be formulated with the method of Lagrange multipliers.

References

[1] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, pages 1141–1151, 2017.

[2] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. Density modeling of images using a generalized normalization transformation. In *International Conference on Learning Representations*, 2016.

[3] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.

[4] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[5] Fabrice Bellard. BPG image format. https://bellard.org/bpg, 2014.

[6] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Universal deep neural network compression. In *NeurIPS Workshop on Compact Deep Neural Network Representation with Industrial Applications (CDNNRIA)*, 2018.

[7] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

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

[9] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4385–4393, 2018.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[11] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[12] Eastman Kodak. Kodak lossless true color image suite (PhotoCD PCD0992). http://r0k.us/graphics/kodak, 1993.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[14] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. In *International Conference on Learning Representations*, 2019.

[15] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4394–4402, 2018.

[16] David Minnen, Johannes Ballé, and George D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10794–10803, 2018.

[17] Antonio Ortega and Kannan Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(6):23–50, 1998.

[18] Majid Rabbani. JPEG2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 11(2):286, 2002.

[19] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *Proceedings of the International Conference on Machine Learning*, pages 2922–2930, 2017.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[21] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

[22] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

[23] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations*, 2017.

[24] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5306–5314, 2017.

[25] Jakub Tomczak and Max Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 2018.

[26] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

[27] Gregory K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

[28] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems & Computers*, volume 2, pages 1398–1402, 2003.

[29] Ram Zamir and Meir Feder. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory*, 38(2):428–436, 1992.

[30] Jacob Ziv. On universal quantization. *IEEE Transactions on Information Theory*, 31(3):344–347, 1985.

Appendices

A. Comparison of our refined probabilistic model to [16]

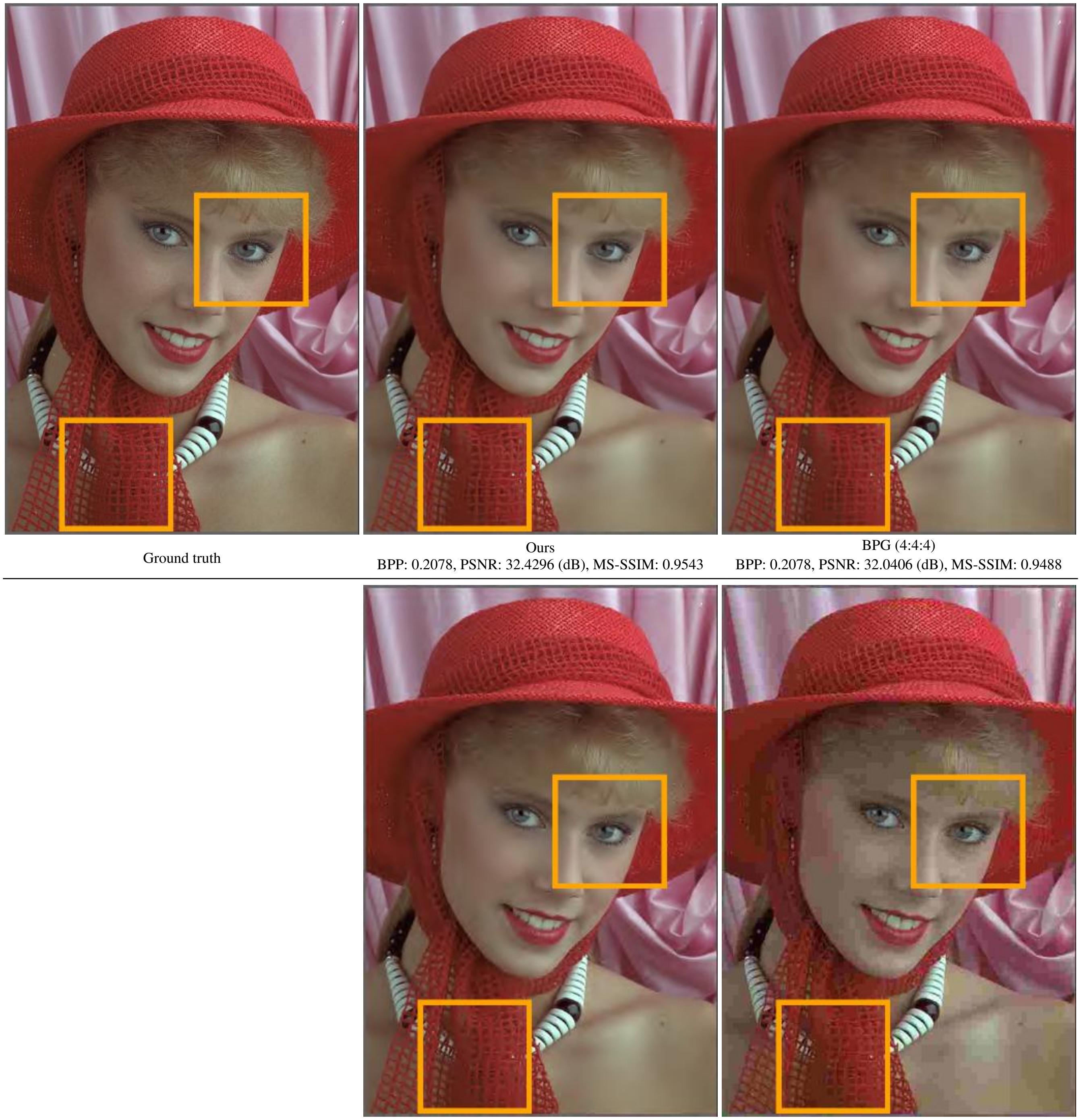
The major difference from [16] is the conditioning part of λ, Δ. Furthermore, there are some differences from [16] in the probabilistic model, which we highlight in Table 1 with red color.

Table 1: Comparison of our refined probabilistic model to [16].

Probability	Modeling in [16]	Modeling in ours
$p_\phi(\mathbf{w}, \mathbf{z} \vert \mathbf{x})$	$p_\phi(\mathbf{z} \vert \mathbf{x}) p_\phi(\mathbf{w} \vert \mathbf{z})$	$p_\phi(\mathbf{z} \vert \mathbf{x}, \lambda, \Delta) p_\phi(\mathbf{w} \vert \mathbf{z}, \mathbf{x}, \lambda, \Delta)$
$q_\theta(\mathbf{x} \vert \mathbf{w}, \mathbf{z})$	$\delta(\mathbf{x} - g_\theta(\mathbf{z}))$	$\delta(\mathbf{x} - g_\theta(\mathbf{z}, \mathbf{w}, \lambda))$
$q_\theta(\mathbf{z} \vert \mathbf{w})$	$\prod_i q_\theta(z_i \vert z_{<i}, \mathbf{w})$	$\prod_i q_\theta(z_i \vert z_{<i}, \mathbf{w}, \lambda, \Delta)$
$q_\theta(\mathbf{w})$	$\prod_i q_\theta(w_i)$	$\prod_i q_\theta(w_i \vert w_{<i}, \lambda, \Delta)$

B. More example images

As supplementary materials, we provide more example images produced by our variable-rate image compression network that is optimized for the MSE loss. We compare our method to the classical image compression codecs, i.e., JPEG, JPEG2000, and BPG. We adapt and match the compression rate of our variable-rate network to the rate of BPG by adjusting the Lagrange multiplier λ and the quantization bin size Δ. All the examples show that our method outperforms the state-of-the-art BPG codec in both PSNR and MS-SSIM measures at the same bits per pixel (BPP). Visually, our method provides better quality with less artifacts than the classical image compression codecs. We put orange boxes to highlight the visual differences in Figure 11 and Figure 13, and the orange-boxed areas are magnified in Figure 12 and Figure 14, respectively.



Ground truth

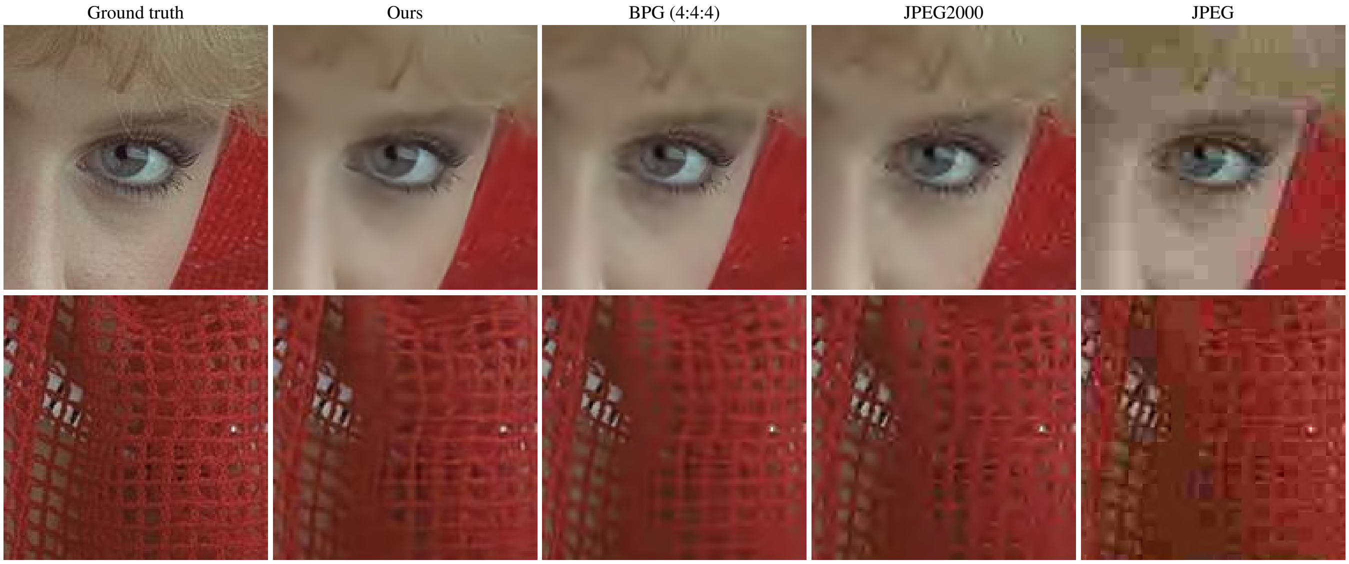
Ours
BPP: 0.2078, PSNR: 32.4296 (dB), MS-SSIM: 0.9543

BPG (4:4:4)
BPP: 0.2078, PSNR: 32.0406 (dB), MS-SSIM: 0.9488

JPEG2000
BPP: 0.2092, PSNR: 30.9488 (dB), MS-SSIM: 0.9342

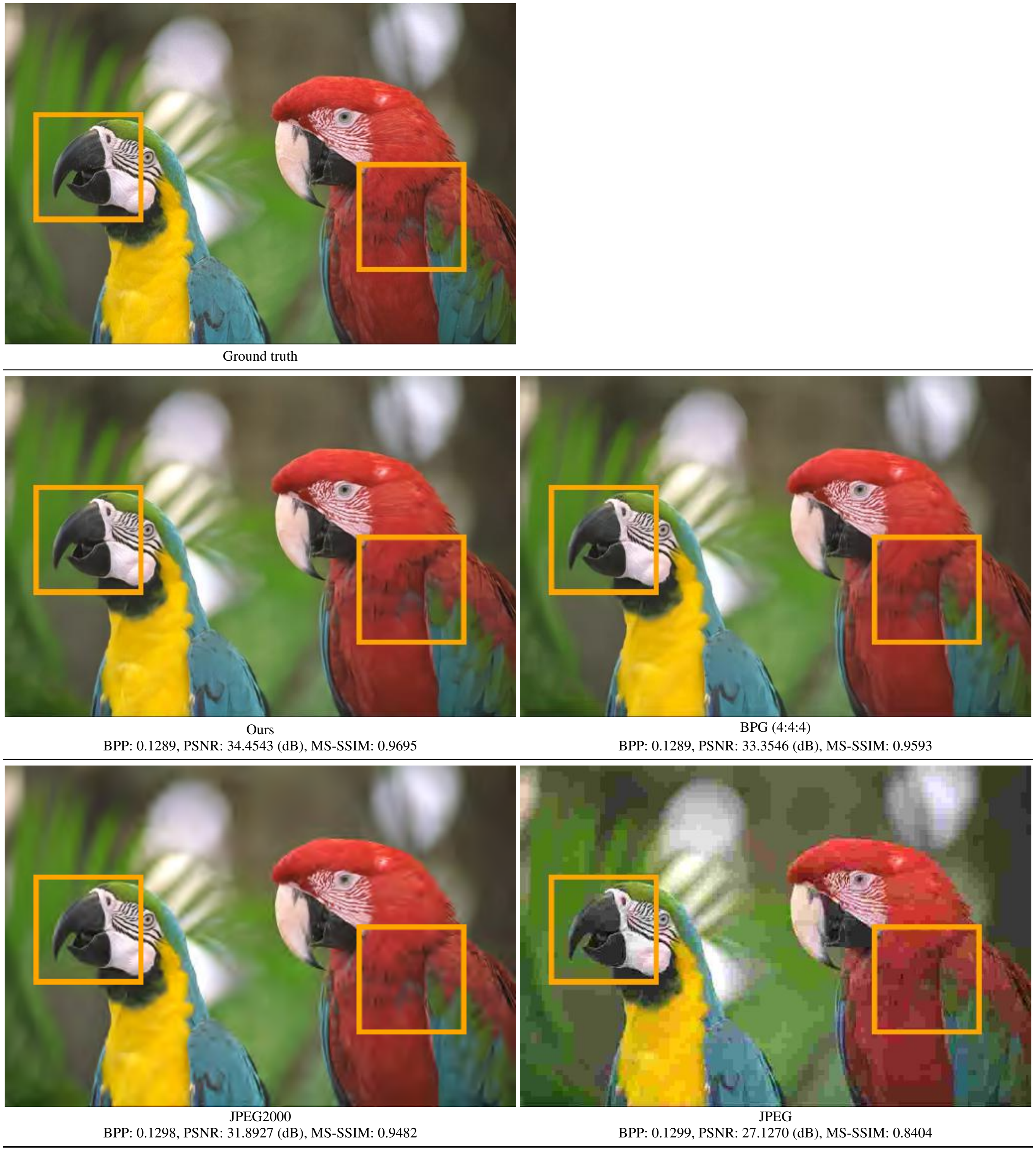
JPEG
BPP: 0.2098, PSNR: 28.1758 (dB), MS-SSIM: 0.8777

Figure 11: PSNR, MS-SSIM, and visual quality comparison of our variable-rate deep image compression method and classical image compression algorithms (BPG, JPEG2000, and JPEG) for the Kodak image 04. Our method outperforms the state-of-the-art BPG codec in both PSNR and MS-SSIM measures. We put orange boxes to highlight the visual differences.



Bits per pixel (BPP)	0.2078	0.2078	0.2092	0.2098
PSNR (dB)	32.4296	32.0406	30.9488	28.1758
MS-SSIM	0.9543	0.9488	0.9342	0.8777

Figure 12: Visual quality comparison of our variable-rate deep image compression method and classical image compression algorithms (BPG, JPEG2000, and JPEG) for the Kodak image 04 in the orange-boxed areas of Figure 11.



Ground truth

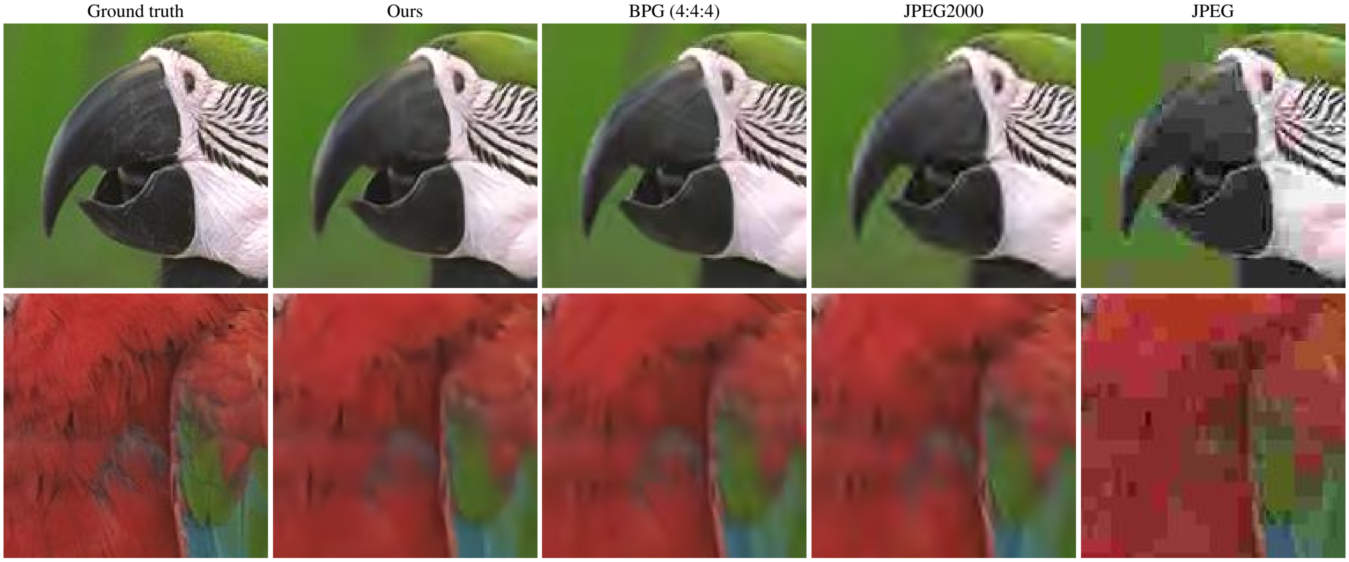
Ours
BPP: 0.1289, PSNR: 34.4543 (dB), MS-SSIM: 0.9695

BPG (4:4:4)
BPP: 0.1289, PSNR: 33.3546 (dB), MS-SSIM: 0.9593

JPEG2000
BPP: 0.1298, PSNR: 31.8927 (dB), MS-SSIM: 0.9482

JPEG
BPP: 0.1299, PSNR: 27.1270 (dB), MS-SSIM: 0.8404

Figure 13: PSNR, MS-SSIM, and visual quality comparison of our variable-rate deep image compression method and classical image compression algorithms (BPG, JPEG2000, and JPEG) for the Kodak image 23. Our method outperforms the state-of-the-art BPG codec in both PSNR and MS-SSIM measures. We put orange boxes to highlight the visual differences.



Ground truth Ours BPG (4:4:4) JPEG2000 JPEG

Bits per pixel (BPP) PSNR (dB) MS-SSIM	0.1289 34.4543 0.9695	0.1289 33.3546 0.9593	0.1298 31.8927 0.9482	0.1299 27.1270 0.8404

Figure 14: Visual quality comparison of our variable-rate deep image compression method and classical image compression algorithms (BPG, JPEG2000, and JPEG) for the Kodak image 23 in the orange-boxed areas of Figure 13.