

# Asymmetric Gained Deep Image Compression With Continuous Rate Adaptation

Ze Cui<sup>1</sup>, Jing Wang<sup>1</sup>, Shangyin Gao<sup>1</sup>, Tiansheng Guo<sup>1</sup>, Yihui Feng<sup>1</sup>, Bo Bai<sup>1</sup>

<sup>1</sup>Huawei Technologies, Beijing, China

{cuize1, wangjing215, gaoshangyin, guotiansheng1, fengyihui, baibo3}@huawei.com

## Abstract

With the development of deep learning techniques, the combination of deep learning with image compression has drawn lots of attention. Recently, learned image compression methods had exceeded their classical counterparts in terms of rate-distortion performance. However, continuous rate adaptation remains an open question. Some learned image compression methods use multiple networks for multiple rates, while others use one single model at the expense of computational complexity increase and performance degradation. In this paper, we propose a continuously rate adjustable learned image compression framework, Asymmetric Gained Variational Autoencoder (AG-VAE). AG-VAE utilizes a pair of gain units to achieve discrete rate adaptation in one single model with a negligible additional computation. Then, by using exponential interpolation, continuous rate adaptation is achieved without compromising performance. Besides, we propose the asymmetric Gaussian entropy model for more accurate entropy estimation. Exhaustive experiments show that our method achieves comparable quantitative performance with SOTA learned image compression methods and better qualitative performance than classical image codecs. In the ablation study, we confirm the usefulness and superiority of gain units and the asymmetric Gaussian entropy model.

## 1. Introduction

Image compression is one of the most fundamental and valuable problems in image processing and computer vision. In the last decades, many researchers have worked for the development and optimization of the classical image compression codecs, such as JPEG [32], JPEG2000 [27] and BPG [8]. To remove redundancy within images, basic modules of the classical codes, including transform coding, entropy coding and quantization, have been sophisticatedly designed and applied. Since these modules are artificially designed and optimized separately, it is not easy to obtain

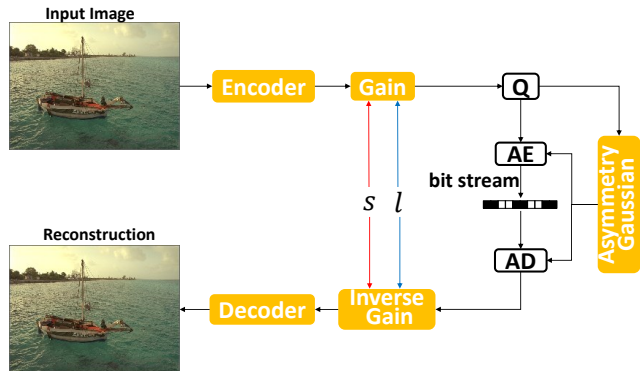


Figure 1. AG-VAE framework. We achieve rate adaptation by inserting a gain unit after encoder and an inverse gain unit before decoder. The bit rate could be adjusted continuously with the change of the gain vector index  $s$  and the interpolation coefficient  $l$ . The asymmetric Gaussian entropy model estimates entropy of the gained and quantized latent representation accurately. an optimal solution for different evaluation indicators.

Recently, deep learning has achieved significant breakthroughs in various learning problems such as image classification, object detection. Thanks to variational autoencoder (VAE) and scalar quantization assumption [5, 4, 6, 25, 28, 1], end-to-end learned image compression framework also has derived significant interests. With the optimization of entropy estimation modules [7, 20, 16] and autoencoder network structure [26, 3, 17, 19, 31, 42, 18, 9], the VAE-based image compression methods have achieved better rate-distortion (R-D) performance than the classical image compression codecs on common metrics, such as Peak Signal-to-Noise Ratio (PSNR) and Multi-Scale-Structural Similarity Index measure (MS-SSIM) [33].

Some VAE-based image compression methods need to train multiple fixed-rate models to realize rate adaption, each model for one rate. Therefore, the training cost and memory requirement increase dramatically with the growth and refinement of the desired rate range. Instead of using multiply models, some other methods achieve the rate adaptation using one single model. The RNN-based schemes [29, 30, 13] encode the input image progressively, but they suffer from bad R-D performance. The conditional autoen-

coder [10, 34] incorporates fully connected layers into the convolution unit to achieve discrete rate adaptation while increasing the network’s computational complexity and memory requirement. Mixed bin sizes [10] are introduced to extend the range coverage from finite discrete points to a broad rate range, but they induce R-D performance degradation. Bottleneck scaling scheme [28, 2] ignores compatibility between autoencoder and scaling parameters and has a poor performance in low bit rate range. Although providing feasible solutions to rate adaptation in a single model, the methods mentioned above have various practical problems such as performance degradation, computational complexity increase and memory increase.

As shown in Figure 1, we propose a novel image compression framework, AG-VAE. It can continuously adjust the bit rate in one single model and achieves comparable R-D performance with SOTA learned image compression methods in quantitative metrics and qualitative visual quality. Based on the unevenness of channel redundancy, we design a plug-and-play variable-rate block, gain unit. By simple channel-wise multiplication, the gain unit rescales the latent representation. The degree of information loss is then controlled in the quantization process.

We address two critical challenges of the proposed framework. First, the inverse-gain unit is introduced to avoid performance degradation. Second, we study the reconstruction assumption of gain units to deduce the exponent interpolation formula, enabling continuous rate adaptation without extra training. To avoid entropy estimation error for the samples with the asymmetric distribution, we also introduce the asymmetric Gaussian entropy model to achieve good R-D performance. To demonstrate the universality of gain units, we further integrate them into other backbone architectures [7, 20]. Besides, we also compare gain units with previous rate adaptation methods from additional computation and performance degradation.

Our contributions can be summarized as follows:

- We introduce gain units to achieve discrete rate adaptation in one single model. With negligible additional computational cost, our method has a similar performance with SOTA learned image compression method.
- We propose the exponent interpolation, which can generate gain vectors at the arbitrary bit rate. The exponent interpolation formula extends the rate’s coverage from finite discrete points to a broad continuous range without an extra training process.
- Gain units with exponent interpolation can be easily generalized to all VAE-based image compression methods while avoiding performance degradation.
- We propose the asymmetric Gaussian entropy model to achieve more accurate entropy estimation. Less bit rate is required to reach the same distortion level.

## 2. Related Works

**Learned Image Compression.** The VAE-based framework could be counted as a nonlinear transforming coding model [5, 6]. The transforming process could be mainly divided into four parts: The encoder that maps an image  $x$  into a latent representation,  $y = f_\theta(x)$ ; The quantizer that transforms the latent representation into the discrete values,  $\hat{y} = Q(y)$ ; The entropy model that estimates the distribution of  $\hat{y}$  to get the minimum rate achievable with lossless entropy source coding [11],  $R_\varphi(\hat{y})$ ; And the decoder that transforms the quantized latent representation to the image,  $\hat{x} = g_\phi(\hat{y})$ . The entire framework can be trained jointly by optimizing the following loss function as:

$$\min_{\theta, \phi, \varphi} R_\varphi(Q(f_\theta(x))) + \beta \cdot D(x, g_\phi(Q(f_\theta(x)))) \quad (1)$$

where  $R_\varphi(\cdot)$  represents the expected code length (bit rate) of the quantized latent representation and  $D(\cdot)$  measures the distortion between the input image and the reconstructed image. The Lagrange multiplier  $\beta$  is a constant in the training process to specify the R-D tradeoff of the trained model [24]. Therefore, the VAE-based image compression methods need to use multiple fixed-rate models trained under different  $\beta$  to adjust the different compression performance of images. However, the multi-model scheme only realizes variable rates in several discrete points of the R-D curve, while memory consumption increases proportionally.

**Rate Adaptation Methods.** The first variable-rate learned image compression was proposed by Toderici et al. [29]. Instead of autoencoder structure, they adopt convolutional LSTM networks. The network is only trained once and can progressively transmit bits. The more bits are sent, the more accurate the image reconstruction is. Subsequently, the LSTM-based scheme was widely adopted, and new techniques were absorbed in it, such as residual scale reconstruction, better entropy coding, and spatial adaptive bit rates [30, 13]. However, the LSTM-based schemes can not outperform JPEG2000 [27] in terms of R-D performance and can not achieve continuous variable-rate adaptation. The LSTM network needs to be inferred multiple times to reconstruct a high-quality image, which is time-consuming and impractical for real-world applications.

Choi et al. [10] proposed a variable rate image compression framework with a conditional autoencoder, which incorporated fully connected networks into the convolution unit and adjusted compression performance with the Lagrange multiplier. Mixed bin sizes were introduced to control quantization loss and finetune bit rate in [10]. However, additional fully connected layers of the conditional convolution increase the computational complexity and memory of the network. Besides, the adjustment of the bin size influences the R-D performance to some extent. It also causes the dilemma of selecting the best combination of the Lagrange multiplier and quantization size in the intersection

of adjacent coarse-adjusting coverages. Yang et al. [34] proposed a modulated autoencoder to realize rate adaptation in several discrete points of the R-D curve. Similar to conditional autoencoder [10], the modulated network introduced fully-connected layers into the autoencoder, which also caused the increase of memory and computation.

Thesis et al. [28] first trained the autoencoder networks at a high bit rate. The pre-trained autoencoder was then fixed and incorporated with scale parameters to achieve rate adaptation. Nevertheless, the incompatibility between autoencoder and scaling parameters led to performance degradation, especially in the R-D curve’s low-rate segmentations. Akabari et al. [2] proposed a stochastic rounding-based quantization scheme and replaced the loss term with rate estimation of the loss function to enable a single model to operate at different bit rates. However, the alteration of loss function made its performance in PSNR much lower than BPG [8]. Compared to the closely related bottleneck scaling methods [8, 28], the proposed gain units provide more insights on strengthening autoencoder’s suitability, gain unit, and inverse gain unit to avoid R-D performance degradation. Based on the reconstruction assumption, we deduce the exponent interpolation formula to achieve continuous rate adaptation in the whole R-D curve.

By incorporating the proposed gain units and a series of optimization schemes [36, 43, 35, 23], we have participated in the Workshop and Challenge on Learned Image Compression 2020 (CLIC2020) [40] and achieved good performance for low bit-rate image compression task [12]. In this paper, we will introduce the motivations and principles of the gain units with exponential interpolation in detail, which can be easily generalized to all VAE-based image compression methods to achieve continuous rate adaptation.

**Entropy Estimation Model.** To obtain accurate entropy estimation of the latent representation, Ballé et al. [7] firstly proposed a zero-mean to model the latent representation. Subsequently, Minnen et al. [7] proposed to estimate the distribution of the latent representation and hyperprior with a mean-and-scale Gaussian entropy model and a non-parametric, fully factorized density model respectively to get the minimum rate achievable with lossless entropy source coding [11], which was still in use by the current learned image compressions methods [20, 10, 19, 42]. However, the symmetric Gaussian entropy model has insufficient degrees of freedom and may induce large estimation errors for natural images with other distributions.

### 3. Proposed Method

In this section, we present our proposed image compression framework AG-VAE, as shown in Figure 1. First, we introduce the principles of the gain units for discrete rate adaption. Then, we depict how the exponent interpolation formula enables the gain units to achieve continuous rate

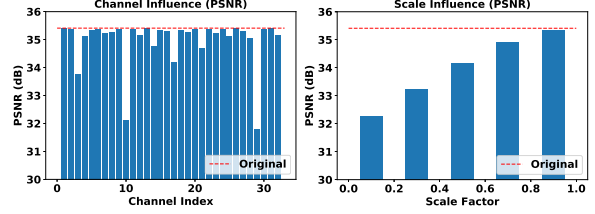


Figure 2. Illustration of channel influences on reconstruction distortion. Left: PSNR degradation of each channel(channel by channel, the first 32 channels of the quantized feature map). Right: PSNR degradation of one channel with various scale factors.

adaption. Furthermore, we extend the gain unit to hyperprior to save more bit rates. Finally, we discuss the effectiveness of the asymmetric Gaussian entropy model.

#### 3.1. Gain Unit

Here, we first conduct a simple experiment to show the channel-wise uneven redundancy in latent representation that widely exists in VAE-based learned image compression frameworks. We taken one image as input of the encoder to obtain its latent representation, denoted as  $y \in R^{c \times h \times w}$ , where  $c, h, w$  represents the number of channels, height, and width of the latent representation respectively. Each channel of  $y$  can be denoted as  $y_i \in R^{h \times w}$ , where  $i = 0, 1, \dots, c - 1$  (If not explicitly mentioned,  $c$  is 192 in our framework). We take *kodim20* from Kodak dataset as an example and set the first 32 channels of the latent representation to zero individually. The modified latent representation is then converted back to the RGB domain, and the PSNR degradation of the reconstruction by the absence of different channels is shown in the left part of Figure 2. The Channel-29 is selected as an example due to the worst degradation in the absence experiment, and the corresponding PSNR of the reconstruction under different scale factors is depicted in the right part of Figure 2. With the decrease of the scaling factor, the quality of the reconstruction is also reduced. We can conclude that the channels’ importance varies and can be scaled to control the reconstruction quality. However, lots of the learned image compression methods ignore the uneven redundancy between channels and treat them equally in the quantization process[37].

To fully utilize the above-mentioned property and scale the latent representation flexibly, we design the gain unit. The gain unit is made up of a gain matrix  $M \in R^{c \times n}$ , where  $n$  represents the number of gain vectors. The gain vector can be denoted as  $m_s = \{m_{s,0}, m_{s,1}, \dots, m_{s,c-1}\}$ , where  $s$  represents the index of the gain vectors in the gain matrix. And  $m_{s,i} \in R$  represents the  $i$ th gain value in the gain vector  $m_s$  and  $i$  ranges from 0 to  $c - 1$ . Each channel is associated with its own scale value. The rescale operation of the latent representation is depicted as:

$$\bar{y}_{s,i} = y_i \times m_{s,i} \quad (2)$$

In this way, the quantization loss of the latent represen-

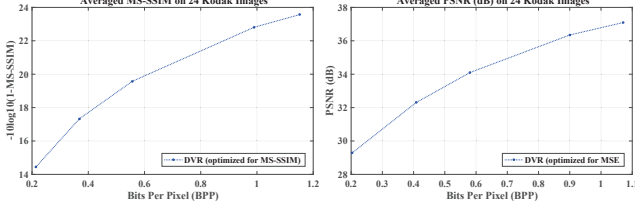


Figure 3. The R-D performance of our DVR method with gain units on 24 Kodak images. In this experiment, we set  $n$  to 5 so that it can produce 5 points of the R-D curve in a single model.

tation can be finely adjusted by the gain vector channel-wisely. Therefore, the network is guided to allocate more bit rates for the channels, which influence the reconstruction quality significantly. The calculation process of the gain unit can be described as:

$$\bar{y}_s = G_\psi(y, s) = y \odot m_s, \quad (3)$$

where  $\bar{y}_s$  is the gained latent representation,  $G_\psi(\cdot)$  represents the gain process, and  $\odot$  represents channel-wise multiplication in Eq 2. What needs to be mentioned is that the gain matrix is trained jointly with the autoencoder network to ensure compatibility between them.

### 3.2. Discrete Variable Rate with Gain Units

In the VAE-based image compression methods, the quantizer is applied element-wisely to round the latent representation  $y$  to the nearest integer. In the Section 3.1, we show that the channel redundancy of the  $y$  is uneven. By scaling the  $y$  to different intervals channel wisely, the gain unit can adjust the channel redundancy, thus control information loss of the quantization process effectively. The quantization process can be formulated as:

$$\hat{y}_s = Q(\bar{y}_s) = \text{round}(\bar{y}_s), \quad (4)$$

where  $\hat{y}_s$  represents the quantized gained latent representation,  $Q(\cdot)$  is the quantization process and  $\text{round}(\cdot)$  denotes element-wisely rounding operation.

Before giving the rescaled and quantized latent representation  $\hat{y}_s$  to the decoder, an inverse rescale operation needs to be done. This operation is used to map  $\hat{y}_s$  back to the same numerical intervals as  $y$ , thus ensuring the reconstruction's correctness. [28] limits the scale and inverse-scale operation to be strictly reciprocal. However, they ignore that the latent representation can not be mapped to the same numerical intervals due to quantization operation by reciprocal inverse scale operation. Here, we adopt another trainable gain unit before the decoder to adaptively rescale  $\hat{y}_s$ , named as the inverse gain unit. Consequently, the gain matrix and gain vector in the inverse-gain unit are denoted as inverse-gain matrix  $M' \in R^{c \times n}$  and inverse gain vector  $m'_s = \{m'_{s,0}, m'_{s,1}, \dots, m'_{s,c-1}\}$ ,  $m'_{s,i} \in R$ . The inverse-gain process can be represented as:

$$y'_s = IG_\tau(\hat{y}_s, s) = \hat{y}_s \odot m'_s, \quad (5)$$

where  $IG_\tau(\cdot)$  represents the inverse gain process. And  $\odot$

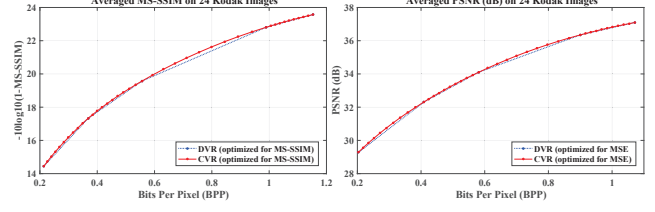


Figure 4. PSNR and MS-SSIM comparison between our DVR method and our CVR method on 24 Kodak images. CVR method owns the same architecture and parameters as DVR method.

represents channel-wise multiplication similar with Eq 2.

The inverse gain vector  $m'_s$  and the corresponding gain vector  $m_s$  always appear in pairs, which could be expressed as  $\{m_s, m'_s\}$ . In the training process, each pair of gain vectors  $\{m_s, m'_s\}$  corresponds to a specific Lagrange multiplier  $\beta_s$  from the predefined finite set of the Lagrange multipliers,  $B \in R^n$ . The gain vector, inverse-gain vector, and Lagrange multiplier are bound together with the subscript  $s$ . Thus, the loss function of the discrete variable rate (DVR) framework is defined as below:

$$\min_{\theta, \phi, \varphi, \psi} \sum_{s=0}^{n-1} R_\varphi(Q(G_\psi(f_\theta(x), s))) + \beta_s \cdot D(x, g_\phi(IG_\tau(Q(G(f_\theta(x), s))), s)), \quad (6)$$

where  $G_\psi(\cdot)$  and  $IG_\tau(\cdot)$  represents the gain process and inverse gain process respectively,  $R_\varphi(\cdot)$  represents the expected bit rate of the quantized gained latent representation.

In the inference process, we change  $s$  to obtain the corresponding gain and inverse-gain vector pair, which could be used to scale the distribution of  $y$  and  $y'_s$  respectively. By this means, we can obtain the desired compression performance limited to several discrete points of the R-D curve. The the R-D curve range depends on the number and value of Lagrange multiplier  $\beta_s \in B$ . We denote the VAE-based image compression method with gain units as the discrete variable rate (DVR) method. It can be seen from Figure 3 that the DVR method could achieve rate adaptation among several discrete points of the R-D curve in a single model.

### 3.3. Exponential Interpolation

Since we use different gain unit pairs to achieve discrete rate adaptation, continuous one can be achieved by interpolation between gain units. In this section, we derive the exponential interpolation based on the property of gain units. The gain unit pair ensures the numerical intervals of  $\hat{y}$  and  $y$  to be the same, which can be formulated as:

$$m_t \cdot m'_t = m_r \cdot m'_r = C, \quad (7)$$

where  $\{m_t, m'_t\}$  and  $\{m_r, m'_r\}$  ( $r, t \in [0, 1, \dots, n-1]$ ) represent the gain vector pairs corresponding to different bit rates, and  $C \in R^c$  is a constant vector. According to Eq 7,



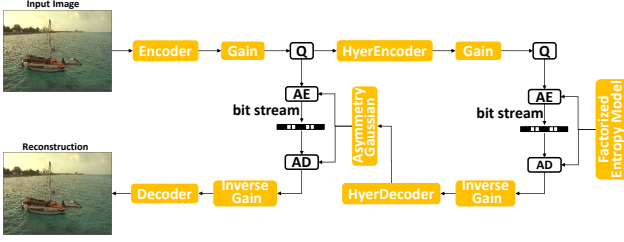


Figure 5. The network architecture of the HCVR. Based on the CVR framework, the HCVR just adds a pair of gain units to the hyperprior autoencoder to obtain flexible entropy estimation. we can derive the exponent interpolation formula as:

$$\begin{aligned} (m_r \cdot m'_r)^l \cdot (m_t \cdot m'_t)^{1-l} &= C, \\ [(m_r)^l \cdot (m_t)^{1-l}] \cdot [(m'_r)^l \cdot (m'_t)^{1-l}] &= C, \quad (8) \\ m_v &= [(m_r)^l \cdot (m_t)^{1-l}], m'_v = [(m'_r)^l \cdot (m'_t)^{1-l}], \end{aligned}$$

where  $\{m_v, m'_v\}$  is the generated gain vector pair and  $l \in R$  is an interpolation coefficient, which controls the corresponding bit rate of the generated gain vector pair. Since  $l$  is a real number, utilizing the exponent interpolation of the gain vector pairs could achieve an arbitrary bit rate between  $t$  and  $r$ . And when  $l$  is equal to 0 or 1, it represents  $\{m_t, m'_t\}$  or  $\{m_r, m'_r\}$  respectively. Without an extra training process and supplementary blocks, we apply the exponent interpolation formula between the adjacent gain vector pairs in the inference process to obtain the Continuously Variable Rate (CVR) method. It could be proved in Figure 4 that the CVR method extends the coverage from finite discrete points to the whole continuous range of the R-D curve while R-D performance not degrades.

### 3.4. Variable Rate of Hyperprior

The hyperprior network [7, 20] can capture the latent representation's spatial dependencies and achieve a more accurate estimation of its distribution. The hyperprior network also adopts the autoencoder structure. It generates the hyperprior latent presentation  $z$ , which is modelled by a non-parametric, fully factorized entropy model.  $z$  also needs to be arithmetically encoded and transmitted, and contributed as a part of the final loss. Therefore, the rate adaption of  $z$  helps to reduce the rate of the learned image compression methods containing the hyperprior network.

The structure of our proposed Hyperprior Continuously Variable Rate (HCVR) method is shown in Figure 5. Another pair of the gain units are introduced into the hyperprior network to scale the hyperprior  $z$ . As the hyperprior  $z$  can be scaled flexibly, the HCVR method reduces the rate consumption of  $z$  without the harm of performance. We will demonstrate the superiority of the HCVR method over the CVR method in the following experiments of Section 4.4.

### 3.5. Gaussian Entropy Model

The matching degree of the parameterized distribution model and real marginal distribution of the latent representation is a significant factor for the expected code length (bit rate) of the quantized latent representation, which decides R-D performance. As the current mainstream model to estimate the distribution of the latent representation, the mean and scale Gaussian entropy model [7] can be formulated as:

$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) \sim N(\mu, \sigma^2), \quad (9)$$

where  $\mu$  and  $\sigma^2$  represent the estimated mean and scale parameters of the latent representation. However, the symmetric Gaussian entropy model has insufficient degrees of freedom and may induce large estimation error for natural images with other distributions. Therefore, we propose the asymmetric Gaussian entropy model [21] as follows:

$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) \sim N(\mu, \sigma_l^2, \sigma_r^2), \quad (10)$$

where  $\sigma_l^2$  and  $\sigma_r^2$  represent the estimated left-scale and right-scale parameter of the latent representation. The asymmetric Gaussian model can achieve better entropy estimation for samples, which do not obey the symmetric gaussian distribution strictly. Besides, all the parameter, including  $\mu$ ,  $\sigma_l^2$  and  $\sigma_r^2$ , are learnable during the training process so that to the extreme that  $\sigma_l^2$  and  $\sigma_r^2$  are the same, the asymmetric Gaussian model could degrade to the symmetric Gaussian model. Therefore, the proposed asymmetric Gaussian entropy model is more flexible and accurate for entropy estimation of the latent representation.

### 3.6. Network Architecture

Our image compression framework AG-VAE is depicted in Figure 6. We adopt the network in [20] as the basic architecture and introduce gain units to realize continuous rate adaptation. The number of channels of the latent representation  $y$  is set to 192 and the kernel size is set to  $3 \times 3$ . The asymmetric Gaussian entropy model is used to replace the symmetric Gaussian entropy model [20] to achieve more accurate entropy estimation. Thus,  $192 \times 3$  channels are required for the mean, left-scale, and right-scale parameters of the asymmetric Gaussian entropy estimation. Besides, we also adopt some optimization methods such as the attention module [36], Universal quantization [43, 35], parallel context models [23], which have been introduced into the deep image compression methods [42, 10, 41], to enhance the R-D performance of the AG-VAE framework.

## 4. Experiments

### 4.1. Implemental Details

**Training** The training set consists of a self-building dataset and a training dataset provided in the CLIC2020 [40]. The self-building dataset contains 5,000 high-quality images collected in various scenes. These images are sampled to

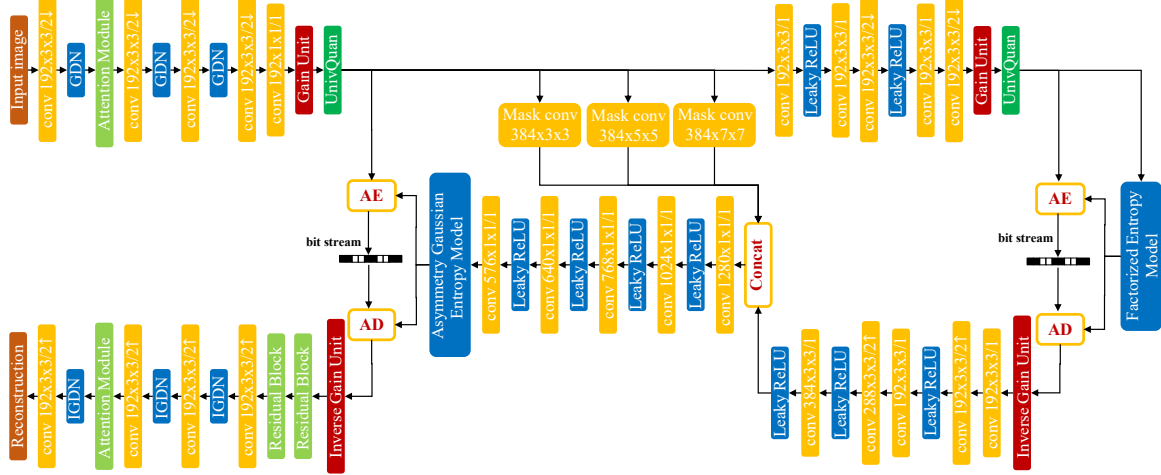


Figure 6. The network architecture of AG-VAE. Convolution parameters are denoted as the number of filters $\times$ kernel height $\times$ kernel width / stride, where  $\uparrow$  and  $\downarrow$  represent upsampling and downsampling respectively. GDN and IGDN represent generalized divisive normalization and the inverse counterpart respectively [4]. Attention Module is used to improve network performance [36]. AE and AD represent the arithmetic encoder and decoder. Masked convolution [23] is utilized to enhance entropy estimation accuracy. The gain unit and inverse gain unit have been interpreted above to achieve rate adaptation. UnivQuant represents universal quantization [43, 35].

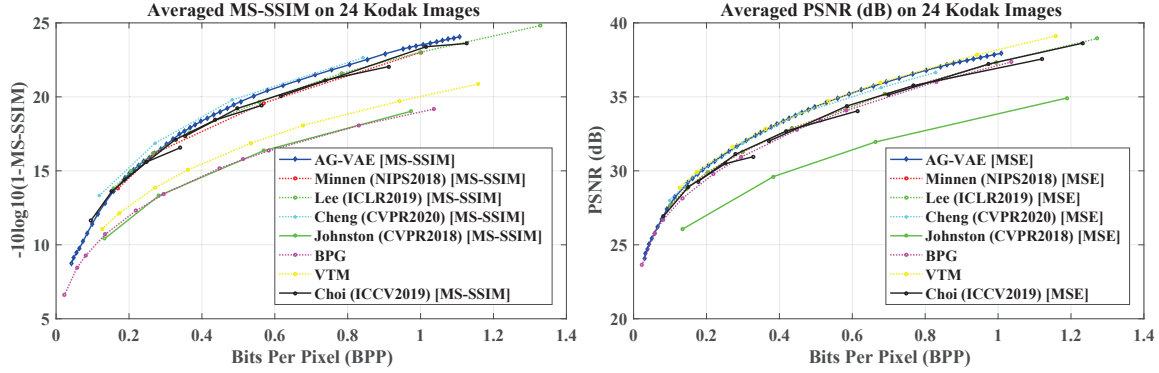


Figure 7. PSNR and MS-SSIM comparison between our variable-rate model AG-VAE and the state-of-the-art image compression methods [16, 20, 9, 10, 13, 8, 38] on 24 Kodak images.

2,000  $\times$  2,000 pixels and saved as lossless PNGs to avoid compression artifacts. We extract two million patches from these downsampled images with a size of 256  $\times$  256 to train the network. We train the model with Adam optimizer [14] for 12 epochs, where the batch size is set to 8, and the learning rate is initially set to  $10^{-4}$  and reduced to half at the 6th epoch. In our experiments,  $n$  denotes the number of gain vector pairs jointly trained with the AG-VAE framework, which is the same as the number of Lagrange multipliers. We prepare two sets of Lagrange multipliers  $B_{mssim} = \{0.07, 0.03, 0.007, 0.003, 0.001, 0.0006\}$  and  $B_{mse} = \{0.05, 0.03, 0.007, 0.003, 0.001, 0.0003\}$ , which correspond to the models trained with MS-SSIM and MSE loss respectively. In the training process, we randomly select  $s$  from 1 to 6 in each iteration to obtain the gain vector  $m_s$ , inverse-gain vector  $m'_s$  and Lagrange multiplier  $\beta_s$  from gain matrix  $M$ , inverse-gain matrix  $M'$  and  $B_{mssim/mse}$ . The selected gain/inverse-gain vector will be optimized jointly with the entire framework under the corresponding Lagrange multiplier.

**Inference** Given the target image and the target rate, we can obtain large-scale discrete rate adaptation by selecting the index  $s$ , while adjusting the interpolation coefficient  $l$  to achieve fine continuous rate adaptation. The bit rate increases as the values of  $s$  and  $l$  increase. When  $l$  is equal to 0 or 1, the discrete rate at  $s$  or  $s + 1$  can be achieved. In practical use, the parameters  $s$  and  $l$  are also arithmetically encoded and decoded along with the latent representation.

## 4.2. Performance Comparison

**Rate-distortion Performance** As shown in Figure 7, We compare the performance of our variable-rate framework AG-VAE to the state-of-the-art learned image compression methods [16, 20, 9] deploying multiple fixed-rate models, the variable-rate learned image compression methods [10, 13], and the classical image compression codecs [8, 38] on the Kodak dataset [15]. The results optimized by MSE or MS-SSIM are presented in two separate plots. With a single model, AG-VAE achieves better R-D performance than those of multiple-networks methods [16, 20, 9] in PSNR,

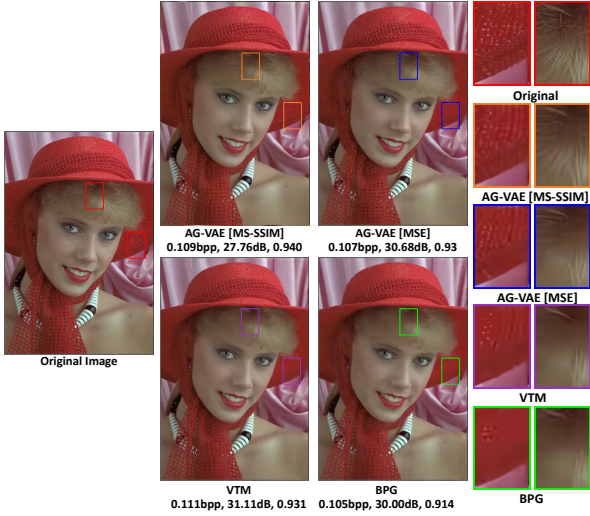


Figure 8. Visualization comparison of reconstructed images *kodim04* from Kodak dataset with approximately 0.1 bpp.

which are believed to be the state-of-the-art ANN-based approach. In MS-SSIM, AG-VAE obtains comparable R-D performance with that of Cheng et al. [9] and even better R-D performance than multiple-networks methods [16, 20]. Compared with other variable-rate learned image compression methods [13, 10], AG-VAE achieves better R-D performance and adjusts rate flexibly while avoiding performance degradation. In particular, the AG-VAE obtains better results than the widely used classical image codec BPG [8] and yields competitive results with VTM [38] in PSNR, which is considered to be the best intra-frame encoding methods of the next-generation compression standard Versatile Video Coding (VVC) [22].

**Visual Results** Figure 8 shows the reconstructed images *kodim04* with approximately 0.10 bpp from the Kodak dataset [15], which are generated from the AG-VAE methods and classical image compression codecs [8, 38] to assess qualitative performance. We observe that the classical codecs suffer from blurring artifacts [8, 38]. In contrast, the proposed AG-VAE optimized by MSE or MS-SSIM recover more details and alleviate the blurring artifacts better. More qualitative results are included in supplementary materials.

### 4.3. Comparison of Variable-Rate Methods

**Rate-distortion Performance** To demonstrate the superiority of gain units, we incorporate the proposed HCVR method or previous rate adaptation methods [28, 10] into the mainstream VAE-based image compression architecture [20]. We also compare those methods with the RNN-based variable-rate image compression method [13]. From Figure 9, we can find that the proposed HCVR method could adjust bit rate flexibly and maintain good performance in the whole range of the R-D curve. However, the method [10] suffers from performance degradation in high-rate segmentations of the R-D curve and intersection of different

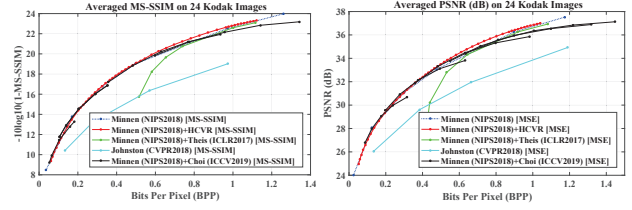


Figure 9. PSNR and MS-SSIM comparison between the proposed HCVR method, [10], [28] and the corresponding method [20] with multiple fixed-rate models on 24 Kodak images.

Basic Method	HCVR		Thesis et al. [28]		Choi et al. [10]	
	Para.	FLOPs	Para.	FLOPs	Para.	FLOPs
Ballé et al. [7]	0.076	0.0004	0.023	0.0004	0.315	0.0428
Minnen et al. [20]	0.040	0.0002	0.010	0.0002	0.140	0.0193

Table 1. The percentages of additional parameters and computation in the proposed HCVR method, the bottleneck-scaling method [28], and the Conditional Conv [10].

bit rate areas of the R-D curve. Meanwhile, the method [28] suffers from bad performance degradation due to the incompatibility between autoencoder and rate-scaling factors. The RNN-based method [13] has far lower R-D performance than counterparts of other methods.

**Additional Computation and Parameters** Parameter and computational quantity are important metrics of whether the learned image compression methods can be popularized and applied. Variable-rate blocks avoid the multiplication of network memory but introduce new computation modules. The RNN-base method achieves better reconstruction quality with iterations, the running time of which increases proportionally. Therefore, We compare the additional parameter percentages Para. and computation percentages FLOPs between the single fix-rate model [7, 20] and other classical variable-rate methods, including the proposed HCVR ( $n_{hp} = 6$ ), the bottleneck-scaling scheme [28], and the Conditional Conv [10]. It can be seen in Table 1 that, compared with the previous classical solution to rate adaptation [10], the additional parameter percentages of our HCVR method and the bottleneck-scaling method [28] are nearly seven times smaller. The additional FLOPs percentages of our HCVR method is nearly 100 times smaller. But the bottleneck-scaling method [28] suffer from severe performance degradation in the low-rate region. Therefore, we can conclude that the proposed HCVR method utilizes the trivial additional parameters and computation to endow the fixed-rate models with continuous rate adaptation while avoiding performance degradation.

### 4.4. Ablation Study

**Generalizability of Gain Unit** Since there is no need to modify the internal structure of the network, gain units can be easily introduced to almost all the VAE-based image compression methods. We verify the performance of gain units on different VAE-based image compression methods,

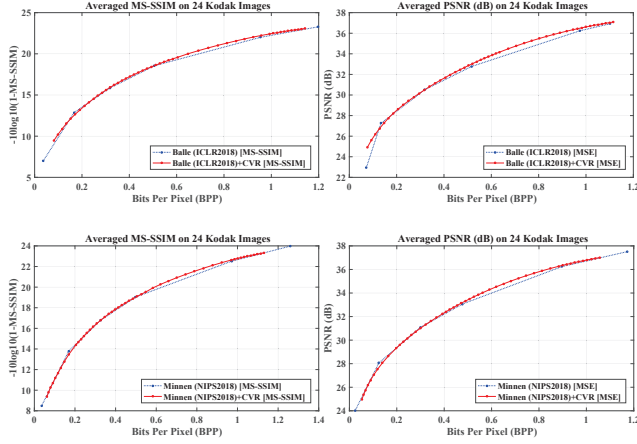


Figure 10. PSNR and MS-SSIM comparison between the proposed HCVR and the corresponding learned image compression methods with multiple fixed-rate models on 24 Kodak images. The basic models in the upper row and the bottom row are the method in [7] and the method in [20] respectively.

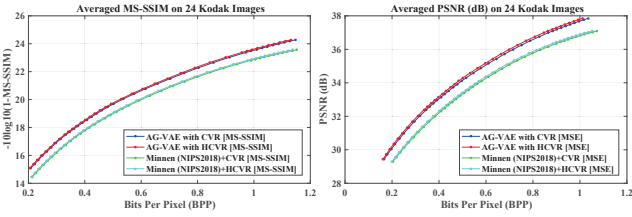


Figure 11. PSNR and MS-SSIM comparison between the HCVR and the CVR on 24 Kodak images. The basic models are the AG-VAE framework proposed in the paper and the learned image compression method proposed by Minnen et al. [20].

including Ballé et al. [7] and Minnen et al. [20]. According to the process introduced in [7, 20], we have reproduced the networks, all of which are trained with different Lagrange multipliers separately to get multiple fixed-rate models in different bit rates. Then, we adopt a single model of methods in [7, 20] as the basic architectures and utilize the CVR method mentioned above to enable them to achieve continuously variable rate in a single model. In Figure 10, we compare our variable-rate networks with corresponding multiple fixed-rate models in PSNR and MS-SSIM respectively. It could be observed that our variable-rate networks in a single model obtain similar R-D performance with those of the multiple fixed-rate models individually optimized for several discrete fixed Lagrange multipliers. Besides, our variable-rate networks based on the basic architectures [7, 20] also utilize the exponent interpolation formula to achieve continuous rate adaptation in a single model, like the proposed AG-VAE framework.

**HCVR Method** By adjusting the bit rate of hyperprior  $z$  flexibly, the HCVR method could achieve more accurate entropy estimation for the distribution of the variable-rate latent representation  $\hat{y}_s$ . We adopt the AG-VAE and the image compression method in [20] as the basic frameworks to demonstrate the superiority of the HCVR method over

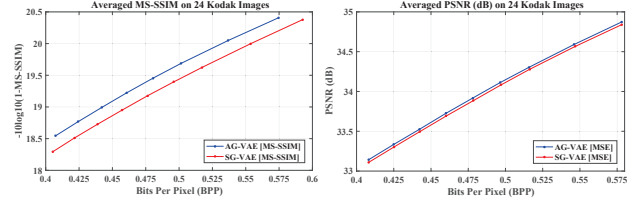


Figure 12. PSNR and MS-SSIM comparison between the AG-VAE and the SG-VAE on 24 Kodak images.

the CVR method. As shown in Figure 11, methods with the HCVR method could achieve slightly better R-D performance than the counterpart of methods with the CVR method in the whole bit-rate range. The results demonstrate the effectiveness of the HCVR methods in the learned image compression methods containing the hyperprior network.

**Asymmetric Gaussian Model** The images in nature don't always follow a symmetrical Gaussian distribution, which is used to realize entropy estimation in the current learned image compression methods. Therefore, we utilize the asymmetric Gaussian entropy model with a high degree of freedom to reduce entropy estimation errors in the learned image compression methods. When the AG-VAE adopts the previous symmetric Gaussian entropy model, we name it as SG-VAE. To show the difference of compression performance clearly, we cut off R-D curves from 0.4 to 0.6 bpp. As shown in Figure 12, AG-VAE achieves better R-D performance than the counterpart of SG-VAE on both metrics.

## 5. Conclusion

We propose a novel continuously variable-rate deep image compression framework AG-VAE, which achieves comparable quantitative performance with the SOTA learned image compression methods and even better qualitative performance than the classical image codecs. By utilizing the unevenness of channel redundancy, we design the gain units to achieve discrete rate adaptation while avoiding performance degradation effectively. We then deduce the exponent interpolation to enable gain units to achieve continuous rate adaptation without extra training or modules. From the aspect of additional computation, additional parameters, and performance degradation, gain units are the state-of-the-art solution to rate adaptation for the VAE-based image compression methods. Experimental results demonstrate the effectiveness and efficiency of the gain units with the exponent interpolation. Besides, the proposed asymmetric Gaussian entropy model achieves flexible entropy estimation for raw images, which can also be extended to other learned image compression methods. We also want to utilize the AG-VAE framework on MindSpore [39], which is a new deep learning computing framework. These works will be finished in the future.



## References

- [1] Eirikur Agustsson, and Michael Tschannen Fabian Mentzer, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. *NIPS*, 2017. 1
- [2] Mohammad Akbari, Jie Liang, Jingning Han, and Chengjie Tu. Learned variable-rate image compression with residual divisive normalization. *ICME*, 2020. 2, 3
- [3] Johannes Ballé. Efficient nonlinear transforms for lossy image compression. *Picture Coding Symposium.*, 2018. 1
- [4] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. Density modeling of images using a generalized normalization transformation. *ICLR*, 2016. 1, 6
- [5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *Picture Coding Symposium*, 2016. 1, 2
- [6] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimization of nonlinear transform codes for perceptual quality. *ICLR*, 2017. 1, 2
- [7] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *ICLR*, 2018. 1, 2, 3, 5, 7, 8
- [8] Fabrice Bellard. Bpg image format. <https://bellard.org/bpg>, 2014. 1, 3, 6, 7
- [9] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *CVPR*, 2020. 1, 6, 7
- [10] Yoojin Choi, Mostafa Elkhamy, and Jungwon Lee. Variable rate deep image compression with a conditional autoencoder. *ICCV*, 2019. 2, 3, 5, 6, 7
- [11] Thomas M. Cover and Joy A. Thomas. Elements of information theory. *John Wiley and Sons*, 2012. 2, 3
- [12] TianSheng Guo, Jing Wang, Ze Cui, Yihui Feng Yunying Ge, and Bo Bai. Variable rate image compression with content adaptive optimization. *CVPRW*, 2020. 3
- [13] 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 network. *CVPR*, 2018. 1, 2, 6, 7
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6
- [15] Eastman Kodak. Kodak lossless true color image suite (photocd pcd0992), 1993. <http://r0k.us/graphics/kodak>. 6, 7
- [16] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. *ICLR*, 2019. 1, 6, 7
- [17] Haojie Liu, Tong Chen, Peiyao Guo, Qiu Shen, Xun Cao, Yao Wang, and Zhan Ma. Non-local attention optimized deep image compression. *CVPR*, 2019. 1
- [18] Haojie Liu, Tong Chen, Qiu Shen, and Zhan Ma. Practical stacked non-local attention modules for image compression. *CVPRW*, 2019. 1
- [19] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. *CVPR*, 2018. 1, 3
- [20] David Minnen, Johannes Ballé, and Toderici George. Joint autoregressive and hierarchical priors for learned image compression. *NIPS*, 2018. 1, 2, 3, 5, 6, 7, 8
- [21] Nafaa Nacereddine, Salvatore Tabbone, Djemel Ziou, and Latifa Hamami. Asymmetric generalized gaussian mixture models and em algorithm for image segmentation. In *ICPR*, 2010. 5
- [22] Jens-Rainer Ohm and Gary J Sullivan. Versatile video coding—towards the next generation of video compression. *Picture Coding Symposium*, 2018. 7
- [23] Aaron Van Den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *NIPS*, 2016. 3, 5, 6
- [24] Antonio Ortega and Kannan Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 1998. 2
- [25] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. *Proceedings of the 34th International Conference on Machine Learning*, 2017. 1
- [26] Tamar Rott Shaham and Tomer Michaeli. Deformation aware image compression. *CVPR*, 2018. 1
- [27] David Taubman and Michael W Marcellin. Jpeg2000 image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 2013. 1, 2
- [28] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *ICLR*, 2017. 1, 2, 3, 4, 7
- [29] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *ICLR*, 2016. 1, 2
- [30] George Toderici, Sean M Omalley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Full resolution image compression with recurrent neural networks. *CVPR*, 2017. 1, 2
- [31] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. Deep generative models for distribution-preserving lossy compression. *NIPS*, 2018. 1
- [32] Gregory K Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 1992. 1
- [33] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. *Asilomar Conference on Signals, Systems Computers.*, 2003. 1
- [34] Fei Yang, Luis Herranz, Joost Van De Weijer, Jose A Iglesias Guitian, Antonio M Lopez, and Mikhail G Mozerov. Variable rate deep image compression with modulated autoencoder. *IEEE Signal Processing Letters*, 2020. 2, 3
- [35] Ram Zamir and Meir Feder. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory*, 1992. 3, 5, 6
- [36] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual nonlocal attention networks for image restoration. *ICLR*, 2019. 3, 5, 6

- [37] Zhisheng Zhong, Hiroaki Akutsu, and Kiyoharu Aizawa. Channel-level variable quantization network for deep image compression. *IJCAI*, 2020. 3
- [38] VVC Official Test Model VTM. [https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware\\_VTM/tags/VTM-7.1](https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-7.1). 6, 7
- [39] MindSpore, 2020. <https://www.mindspore.cn/>. 8
- [40] Workshop and challenge on learned image compression, 2020. <https://www.compression.cc/>. 3, 5
- [41] Jing Zhou, Sihan Wen, Akira Nakagawa, Kimihiko Kazui, and Zhiming Tan. Multi-scale and context-adaptive entropy model for image compression. *CVPRW*, 2019. 5
- [42] Lei Zhou, Zhenhong Sun, Xiangji Wu, and Junmin Wu. End-to-end optimized image compression with attention mechanism. *CVPRW*, 2019. 1, 3, 5
- [43] Jacob Ziv. On universal quantization. *IEEE Transactions on Information Theory*, 1985. 3, 5, 6