Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules

Zhengxue Cheng[1], Heming Sun[2,3], Masaru Takeuchi[2], Jiro Katto[1]

[1] Department of Computer Science and Communication Engineering, Waseda University, Tokyo, Japan

[2] Waseda Research Institute for Science and Engineering, Tokyo, Japan [3] JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama, Japan

{zxcheng@asagi., hemingsun@aoni., masaru-t@aoni., katto@}waseda.jp

Abstract

Image compression is a fundamental research field and many well-known compression standards have been developed for many decades. Recently, learned compression methods exhibit a fast development trend with promising results. However, there is still a performance gap between learned compression algorithms and reigning compression standards, especially in terms of widely used PSNR metric. In this paper, we explore the remaining redundancy of recent learned compression algorithms. We have found accurate entropy models for rate estimation largely affect the optimization of network parameters and thus affect the rate-distortion performance. Therefore, in this paper, we propose to use discretized Gaussian Mixture Likelihoods to parameterize the distributions of latent codes, which can achieve a more accurate and flexible entropy model. Besides, we take advantage of recent attention modules and incorporate them into network architecture to enhance the performance. Experimental results demonstrate our proposed method achieves a state-of-the-art performance compared to existing learned compression methods on both Kodak and high-resolution datasets. To our knowledge our approach is the first work to achieve comparable performance with latest compression standard Versatile Video Coding (VVC) regarding PSNR. More importantly, our approach generates more visually pleasant results when optimized by MS-SSIM.

1. Introduction

Image compression is an important and fundamental research topic in the field of signal processing for many decades to achieve efficient image transmission and storage. Classical image compression standards include JPEG [1], JPEG2000 [2], HEVC/H.265 [3] and ongoing Versatile Video Coding (VVC) [4] which will be the next generation compression standard expected by the end of 2020. Typically they rely on hand-crafted creativity to present module-

Figure 1: Visualization of reconstructed images *Kodim04* from Kodak dataset with approximately 0.1 bpp.

based encoder/decoder (codec) block diagrams. They use fixed transform matrix, intra prediction, quantization, context adaptive arithmetic coders and various de-blocking or loop filters to reduce spatial redundancy to improve the coding efficiency. The standardization of a conventional codec has historically spanned several years. Along with the fast development of new image formats and the proliferation of high-resolution mobile devices, existing image compression standards are not expected to be an optimal and general solutions for all kinds of image contents.

Various approaches has been investigated for end-to-end image compression. Recently, the notable approaches are context-adaptive entropy models for learned image compression [19, 20, 21] to achieve superior performance among all the learned codecs. The work [19] proposed a hyperprior to add additional bits to model the entropy model. The work [20] jointly combine an autoregressive mask convolution and the hyperprior. The work [21] proposed a quite similar idea by considering two types of contexts, bit-consuming contexts (that is, hyperprior) and bit-free contexts (that is, autoregressive model) to realize a context-

adaptive entropy model. Our proposed methods are based on the development of these recent entropy model techniques to further improve the performance.

In this paper, our main contribution is to present a more accurate and flexible entropy model by leveraging discretized Gaussian mixture likelihoods. We visualize the spatial redundancy of compressed codes from recent learned compression techniques. Motivated from it, we propose to use discretized Gaussian mixture likelihoods to parameterize the distributions, which removes remaining redundancy to achieve accurate entropy model, and thus directly lead to fewer required encoding bits. Besides, we adopt a simplified version of attention module into our network architecture. Attention module can make learned models pay more attention to complex regions to improve our coding performance with moderate training complexity.

Experimental results demonstrate our proposed method leads to the state-of-the-art performance on both PSNR and MS-SSIM quality metrics, in comparison with classical image compression standards HEVC, JPEG2000, JPEG and existing deep learning based compression approaches. To our knowledge, we are the first work to reach very close performance with ongoing versatile video coding test model VTM 5.2 with intra profile in terms of PSNR. Moreover, our method produces visually pleasant reconstructed images when optimizing by MS-SSIM as Fig. 1.

2. Related Work

Hand-crafted Compression Existing image compression standards, such as JPEG [1], JPEG2000 [2], HEVC [3] and VVC [4], reply on hand-crafted module design individually. Specifically, these modules include intra prediction, discrete cosine transform or wavelet transform, quantization, and entropy coder such as Huffman coder or content adaptive binary arithmetic coder (CABAC). They design each module with multiple modes and conduct the rate-distortion optimization to determine the best mode. Especially, VVC [4] supports larger coding unit, more prediction modes, more transform types and other coding tools. Besides, along with the development of classical compression algorithms, some hybrid methods have been proposed, by taking advantage of both conventional compression algorithms and latest learned super resolution approaches, such as [23].

Learned Compression Recently, we have seen a great surge of deep learning based image compression approaches utilizing autoencoder architecture [5], which have achieved a great success with promising results. The development of previous learning based compression models have spanned several years and have many related works. In the early stage, some works are proposed to deal with non-differential quantization and rate estimation to make end-to-end training possible, such as [6, 7, 8]. After that, some works focus on the design of network structure, which

is able to extract more compact and efficient latent representation and to reconstruct high-quality images from compressed features. For instance, some works [9, 10, 11] use recurrent neural networks to compress the residual information recursively, but they mainly relied on binary representation at each iteration to achieve scalable coding for image compression. Some approaches [12, 13, 14] use generative models to learn the distribution of images using adversarial training to achieve better subjective quality at extremely low bit rate. Some approaches include a content-weighted strategy [15] or de-correlating different channels using principle component analysis [16], or consider energy compaction [17], or use deep residual units to enhance the network architecture [18]. Recently, several studies investigate the adaptive context model for entropy estimation to navigate the optimization process of neural network parameters to achieve best tradeoff between reconstruction errors and required bits (entropy), including [19, 20, 21, 22]. Entropy estimation techniques have greatly improved the learned compression algorithms, and the most representative methods are hyperprior model and joint model. However, there is still a gap between the estimated distribution and true marginal distribution of latent representation.

Parameterized Model Some image generation studies have investigated several parameterized distribution models. For instance, standard PixelCNN [24] uses full 256-way softmax likelihoods, but they are extremely memory-consuming. To address this problem, PixelCNN++ [25] proposed a discretized logistic mixture likelihoods to achieve faster training. Learned lossless image compression work L3C [26] followed the PixelCNN++ to use a logistic mixture model. These works are basically used to estimate the likelihoods of 8-bit pixel values with fixed range of $[0, 255]$. However, in learned image compression task, few studies explore the effect of parameterized distribution models.

3. Proposed Method

3.1. Formulation of Learned Compression Models

In the transform coding approach [27], image compression can be formulated by (as Fig. 2(a))

$$
\begin{aligned}
\boldsymbol{y} &= g_a(\boldsymbol{x}; \boldsymbol{\phi}) \\
\hat{\boldsymbol{y}} &= Q(\boldsymbol{y}) \\
\hat{\boldsymbol{x}} &= g_s(\hat{\boldsymbol{y}}; \boldsymbol{\theta})
\end{aligned}
\tag{1}
$$

where $\boldsymbol{x}$, $\hat{\boldsymbol{x}}$, $\boldsymbol{y}$, and $\hat{\boldsymbol{y}}$ are raw images, reconstructed images, a latent presentation before quantization, and compressed codes, respectively. $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are optimized parameters of analysis and synthesis transforms. $U|Q$ represents the quantization and entropy coding. During the training, the quantization is approximated by a uniform noise $\mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ to generate noisy codes $\tilde{\boldsymbol{y}}$. During the inference, $U|Q$ represents
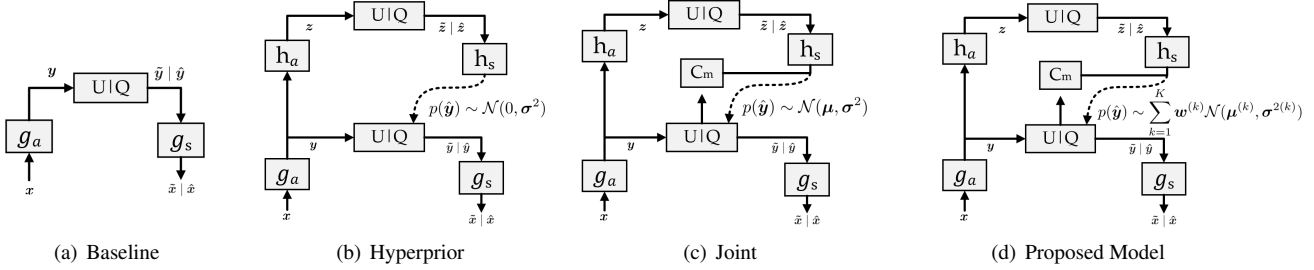
Figure 2: Operational diagrams of learned compression models (a)(b)(c) and proposed Gaussian Mixture Likelihoods (d).

real round-based quantization to generate $\hat{y}$ and followed entropy coders to generate the bitstream. To simplify, we use $\hat{y}$ to denote $\tilde{y}|\hat{y}$. If a probability model $p_{\hat{y}}(\hat{y})$ is given, entropy coding techniques, such as arithmetic coding [28], can losslessly compress the quantized codes. Besides, arithmetic coder are a near-optimal entropy coder, which makes it feasible to use the entropy of y as the rate estimation during the training. In the baseline architecture, the marginal distribution of latent y is unknown and no additional bits are available to estimate $p_{\hat{y}}(\hat{y})$. Typically a non-adaptive density model is used and shared between encoder and decoder, also called factorized prior.

In the work [19], *Ballé* proposed a hyperprior, by introducing a side information z to capture spatial dependencies among the elements of y, formulated by (as Fig. 2(b))

$$z = h_a(y; \phi_h);$$
$$\hat{z} = Q(z) \qquad (2)$$
$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) \leftarrow h_s(\hat{z}; \theta_h)$$

where h_a and h_s denote the analysis and synthesis transforms in the auxiliary autoencoder, where ϕ_h and θ_h are optimized parameters. $p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z})$ are estimated distributions conditioned on $\hat{z}$. For instance, the work [19] parameterized a zero-mean Gaussian distribution with scale parameters $\sigma^2 = h_s(\hat{z}; \theta_h)$ to estimate $p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z})$.

Following that, an enhanced work [20] proposed a more accurate entropy model, which jointly utilize an autoregressive context model (denoted as C_m in Fig. 2(c)) and a mean and scale hyperprior. The work [21] also proposed a similar idea. The operational diagram is explained in Fig. 2(c).

Learned image compression is a Lagrangian multiplier-based rate-distortion optimization. The loss function is

$$\mathcal{L} = \mathcal{R}(\hat{y}) + \mathcal{R}(\hat{z}) + \lambda \cdot \mathcal{D}(x, \hat{x})$$
$$= \mathbb{E}[-\log_2(p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}))] + \mathbb{E}[-\log_2(p_{\hat{z}|\psi}(\hat{z}|\psi))] \quad (3)$$
$$+ \lambda \cdot \mathcal{D}(x, \hat{x})$$

where λ controls the rate-distortion tradeoff. Different λ values are corresponding to different bit rates. $\mathcal{D}(x, \hat{x})$ de-

notes the distortion term. There is no prior for $\hat{z}$, so a factorized density model ψ is used to encode $\hat{z}$ as

$$p_{\hat{z}|\psi}(\hat{z}|\psi) = \prod_i (p_{z_i|\psi}(\psi) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\hat{z}_i) \quad (4)$$

where z_i denotes the i-th element of z, and i specifies to the position of each element or each signal. The remaining part is how to model the $p_{\hat{y}}(\hat{y}|\hat{z})$ accurately.

3.2. Discretized Gaussian Mixture Likelihoods

Whether the parameterized distribution model fits the marginal distribution of $\hat{y}$ is a significant factor for the entropy model, thus affecting rate-distortion performance. Lots of efforts have been conducted to model the conditional probability distribution $p_{\hat{y}}(\hat{y}|\hat{z})$ after decoding $\hat{z}$. *Ballé* work [19] firstly assumed a univariate Gaussian distribution model for the hyperprior, that is,

$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) \sim \mathcal{N}(0, \sigma^2) \quad (5)$$

The improved work [20] extended a scale hyperprior to a mean and scale Gaussian distribution, that is,

$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) \sim \mathcal{N}(\mu, \sigma^2) \quad (6)$$

and combine with a autoregressive model, denoted as *Joint*.

To illustrate how entropy models work, we visualize different entropy models in Fig. 3 with the same network architecture. We only depict the channel with the highest entropy. The first rows visualizes the mean and scale *HyperPrior*. The 1-st column is the quantized codes $\hat{y}$. The second and third columns are the predicted mean μ and scale σ. The 4-th column is normalized values, calculate by $\frac{\hat{y}-\mu}{\sigma}$. It is used to visualize the extent of remaining redundancy which is not captured by entropy models. The 5-th column visualizes the *Required bits* for each element for encoding, which is calculated as $(-\log_2(p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z})))$, which uses predicted distribution models.

Typically, predicted mean μ is close to $\hat{y}$. Complex regions have large σ, requiring more bits for encoding. Instead, smooth regions have smaller σ, resulting to fewer
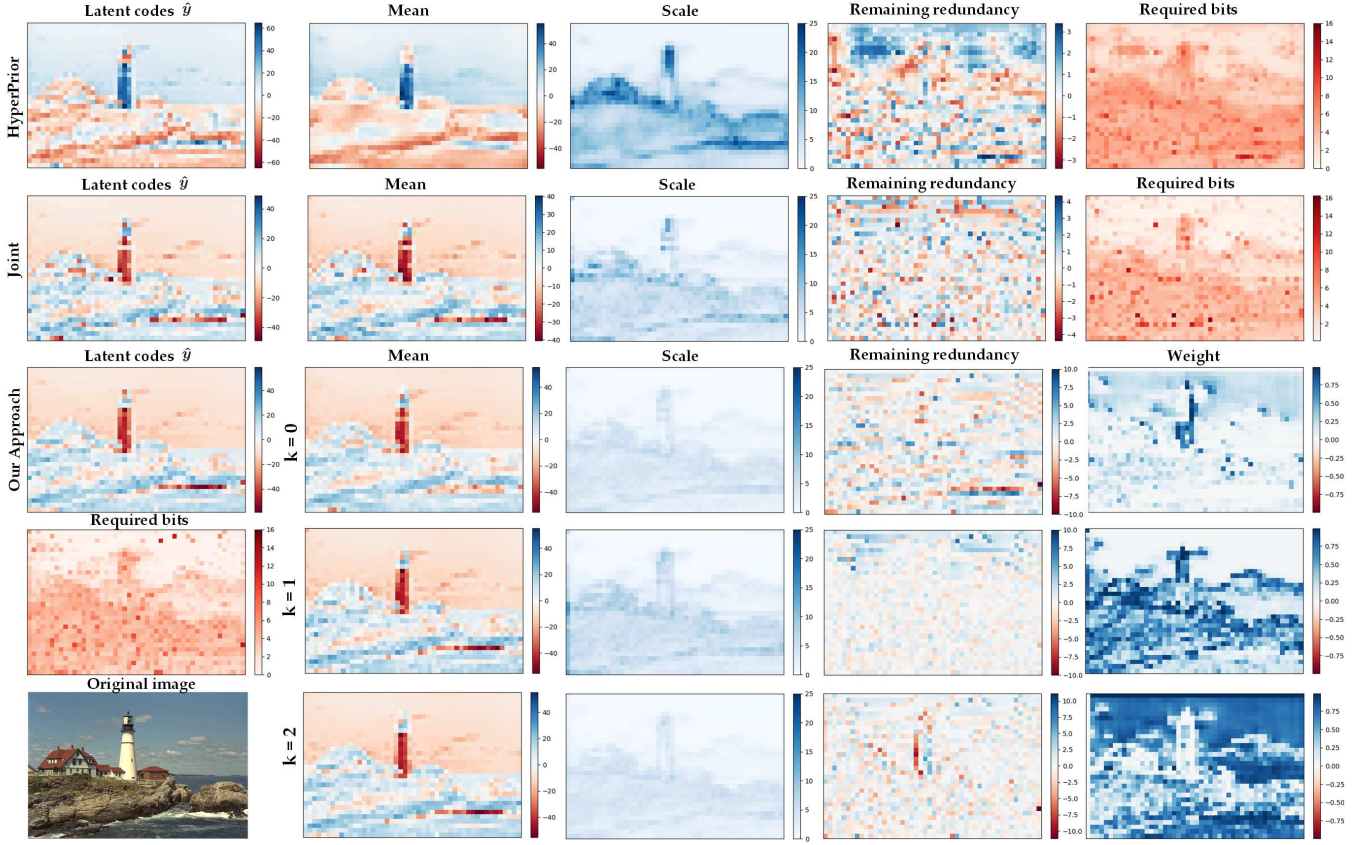
Figure 3: Visualization of different entropy models for the channel with the highest entropy using *kodim21* from Kodak dataset as an example. It shows our approach provides a more flexible parameterized distribution models with smaller scale parameters and better spatial redundancy reduction, which directly results to a more accurate entropy model and fewer bits.

bits for encoding. *HyperPrior* still has spatial redundancy remaining in simple regions, such as sky of *kodim21*. Similarly, we visualize the *Joint* entropy model. Compared with *HyperPrior*, *Joint* removes more structures for normalized values by adding the autoregressive model, which is implemented by a 5×5 mask convolution, to capture the correlation with neighboring elements.

However, *Joint* entropy model is not perfect, because some spatial redundancy is still observed in the 2-nd row, the 4-th column of Fig. 3. Although neighboring elements have already served as the input of context models, parameterized distributions cannot represent well to fully utilize the contexts and information from neighboring elements and additional bits $\hat{z}$. It might be limited by fixed shape of single Gaussian distribution. This motivates us to consider more flexible parameterized models to achieve arbitrary likelihoods. Therefore, we propose the Gaussian mixture model, i.e.

$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) \sim \sum_{k=1}^{K} \boldsymbol{w}^{(k)} \mathcal{N}(\boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{2(k)}) \quad (7)$$

Eq.(7) usually refers to continuous values, but $\hat{y}$ is discrete-valued after quantization. Inspiring by [25], we propose to use discretized Gaussian mixture likelihoods. The reason why we did not use Logistic mixture likelihoods, because Gaussian achieves slightly better performance than logistic [20]. Then the entropy model is formulated as

$$p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) = \prod_i p_{\hat{y}|\hat{z}}(\hat{y}_i|\hat{z})$$

$$p_{\hat{y}|\hat{z}}(\hat{y}_i|\hat{z}) = \left(\sum_{k=1}^{K} w_i^{(k)} \mathcal{N}(\mu_i^{(k)}, \sigma_i^{2(k)}) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2})\right)(\hat{y}_i)$$

$$= c(\hat{y}_i + \frac{1}{2}) - c(\hat{y}_i - \frac{1}{2}) \quad (8)$$

where i specifies the location in feature maps. For example, $\hat{y}_i$ denotes the i-th element of $\boldsymbol{y}$ and μ_i denotes the i-th element of $\boldsymbol{\mu}$. k denotes the index of mixtures. Each mixture is characterized by a Gaussian distribution with 3 parameters, i.e. weights $w_i^{(k)}$, means $\mu_i^{(k)}$ and variances $\sigma_i^{2(k)}$ for each element $\hat{y}_i$. $c(\cdot)$ is the cumulative function. The range
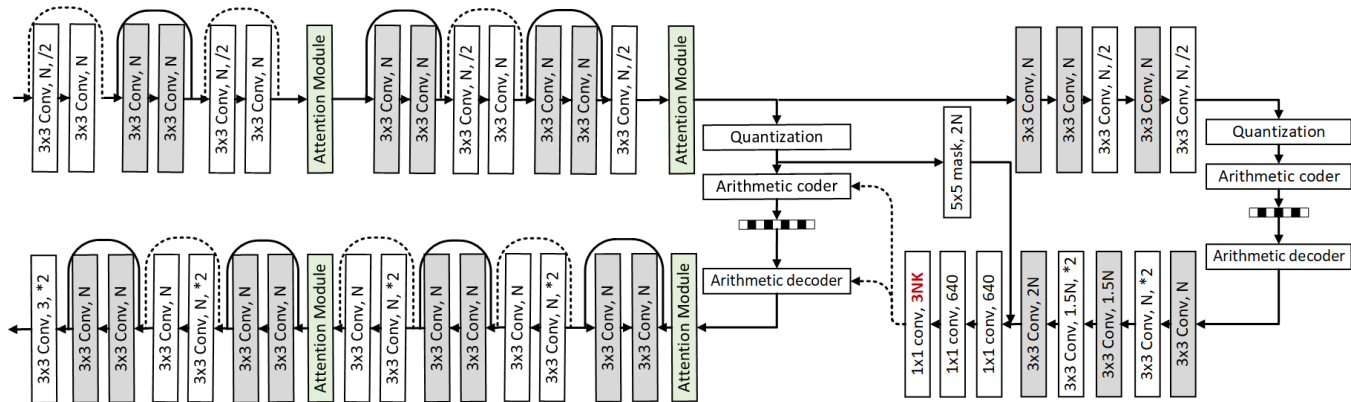
Figure 4: Network architecture.

Table 1: Required bits and quality metrics for *Kodim21*.

Model	PSNR (dB)	MS-SSIM	Rate (bpp)
Joint	33.435	0.980	0.533
Ours	33.623	0.981	0.519

Table 2: Performance of different attention modules.

Module	(a)w/ NLB	(b)w/o NLB	w/o attention
Loss	2.705	2.754	3.026
Time(s)/epoch	1119	336	216

of $\hat{\boldsymbol{y}}$ is automatically learned and unknown ahead of time. To achieve stable training, we clip the range of $\hat{\boldsymbol{y}}$ to [-255, 256] because empirically $\hat{\boldsymbol{y}}$ would not exceed this range. For the edge case of -255, replace $c(\hat{y}_i - \frac{1}{2})$ by zero, i.e. $c(-\infty) = 0$. For the edge case of 256, replace $c(\hat{y}_i + \frac{1}{2})$ by one, i.e. $c(+\infty) = 1$. It provides a numerically stable implementation for training.

The visualization of our approach is shown as the last three rows in Fig. 3. We use $K = 3$ in our experiments. Different from the above two rows, the 5-th column shows the weights $w_i^{(k)}$ for each element and each mixture. Although each mixture still remains some spatial redundancy as shown in the 4-th column in some parts, mixture model can adjust weights to different mixtures and different regions. For instance, the mixture $k = 1$ has some redundancy in the sky as shown in the 4-th row and 4-th column, but the weights of this mixture for the sky are very small. The mixture $k = 2$ remains some redundancy in the while tower as shown in the 5-th row and 4-th column, and mixture model also assigns small weights to the tower regions as shown in the 5-th row and 5-th column. Besides, the scales of our approach are smaller than *Joint* and *Hyperprior*, which demonstrates our entropy model is more accurate, thus directly resulting to fewer bits. *Required bits* at the 5-th row and 1-st column visualizes the required bits of our approach for encoding, which is calculated by $(-\log_2(p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{y}_i|\hat{\boldsymbol{z}})))$. Required bits of our approach is fewer than *Joint* as Table 1 and visualized in Fig. 3. All the models are trained with $\lambda = 0.015$ with the same network architecture.

The Gaussian mixture model can achieve better performance because it selects K most probable values and as-

sign small scale (i.e. high likelihoods) to them. Somehow this mechanism resembles the most probable symbol (MPS) in the Context-based Adaptive Binary Arithmetic coding (CABAC), which has been widely used in traditional video coding standards, but our latent codes are not limited to be binary. To the one extreme, if three selected mean values are the same, it will degrades to a single Gaussian model, which happens in the smooth regions. To the other extreme, three selected mean values are completely different from each other, our model would have three peaks of likelihoods representing three most probable values, which usually happens in the boundaries, edges or complex regions. Besides, all the parameters, weights $w_i^{(k)}$, means $\mu_i^{(k)}$ and variances $\sigma_i^{2(k)}$ for each element, are learnable during the training. Therefore, our method is a flexible and accurate entropy model, and operation diagram is shown as Fig. 2(d).

3.3. Network Architecture

Our network architecture has a similar structure as [18] in Fig. 4. We use residual blocks to increase large receptive field and improve the rate-distortion performance. Decoder side uses subpixel convolution instead of transposed convolution as upsampling units to keep more details. N denotes the number of channels and represents the model capacity. We use Gaussian mixture model, thus requiring $3 \times N \times K$ channels for the output of auxiliary autoencoder.

Furthermore, recent works use attention module to improve the performance for image restoration [29] and compression [30]. The proposed attention module is illustrated in Fig. 5(a), but very time-consuming for training. We simplify this module by removing the non local block, because
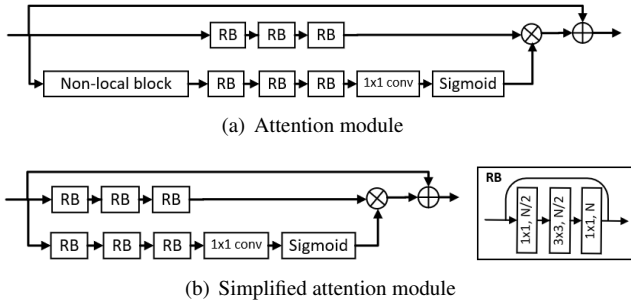
(a) Attention module

(b) Simplified attention module

Figure 5: Different attention modules.

(a) Loss curves with N=128.

(b) RD points with N=128.

(c) Loss curves with N=192.

(d) RD points with N=192.

Figure 6: Ablation Study.

the deep residual blocks can already capture very large receptive field in our network architecture. The loss and training time comparison is given by Table 2, where this loss is trained after 16 epoches. Simplified attention module is shown in Fig. 5(b) and can also reduce the loss with moderate complexity. Attention module can help the networks to pay more attention to challenging parts and reduce the bits of simple parts. Then we insert simplified attention module into encoder-decoder network as Fig. 4.

4. Implementation Details

Training Details   For training, we used a subset of ImageNet database [31], and cropped them into 13830 samples with the size of 256×256. To train our image compression models, the model was optimized using Adam [32] with a batch size of 8. The learning rate was maintained at a fixed value of 1×10^{-4} during the training process, and was reduced to 1×10^{-5} for the last $80k$ iterations.

We optimized our models using two quality metrics, i.e. mean square error (MSE) and MS-SSIM. When optimized by MSE, λ belongs to the set $\{0.0016, 0.0032, 0.0075, 0.015, 0.03, 0.045\}$. N is set as 128 for three lower-rate models, and is set as 192 for three higher rate models. To achieve high subjective quality, we also train the model using MS-SSIM quality metrics [33] and then distortion term is defined by $\mathcal{D}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = 1 - \text{MS-SSIM}(\boldsymbol{x}, \hat{\boldsymbol{x}})$. When optimized by MS-SSIM, λ is in the set $\{3, 12, 40, 120\}$. N is set as 128 for two lower-rate models, and is set as 192 for the two higher-rate models. Each model was trained up to 10^6 iterations for each λ to achieve stable performance.

Evaluation   For comparison, we tested commonly used Kodak lossless image database [34] with 24 uncompressed 768×512 images. To validate the robustness of our proposed method, we also tested our proposed method using CVPR workshop CLIC professional validation dataset [35] with 41 high resolution and high quality images.

To evaluate the rate-distortion performance, the rate is measured by bits per pixel (bpp), and the quality is measured by either PSNR or MS-SSIM, corresponding to op-
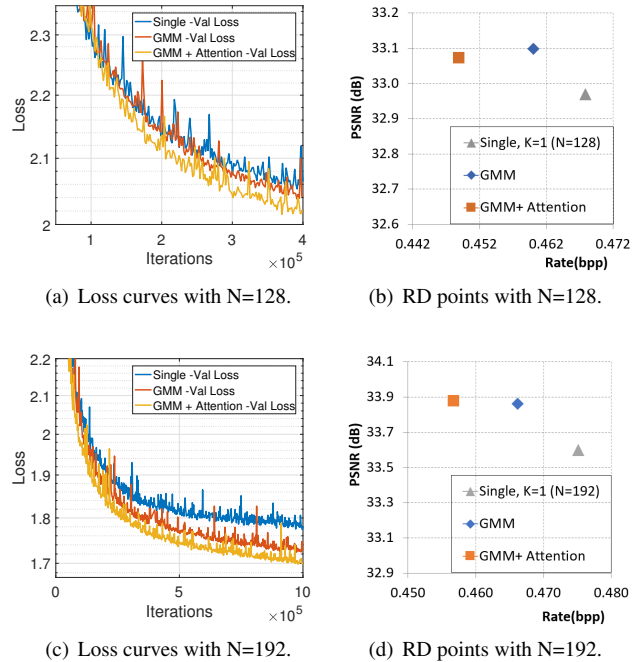
timized distortion metrics. The rate-distortion (RD) curves are drawn to demonstrate their coding efficiency.

Traditional Codecs   For VVC and HEVC, we used the official test model VTM 5.2 (accessed on July 2019) [36] with intra profile and BPG software [37] to test the performance. For both of them, we used non-default *YUV444* format as the configuration, because it prevents the color component loss during color space conversion. We used official test model OpenJPEG [38] with default configuration $YUV420$ to represent the performance of JPEG2000. For JPEG compression, we use PIL library [39].

5. Experiments

5.1. Ablation Study

In order to show the effectiveness of our proposed Gaussian mixture model (GMM) and simplified attention module, we test them with different model capacity, i.e. N=128, and N=192. These models are optimized by MSE with $\lambda = 0.015$. The loss curves are shown in Fig. 6(a) and Fig. 6(c). The asymptotic loss curves shows along with the increase of training iterations, the effectiveness of our proposed approaches become strong. The corresponding rate-distortion points are depicted in Fig. 6(b) and Fig. 6(d), to show the coding gain of proposed approaches. It can be observed our proposed approaches can improve the rate-distortion performance regardless of the model capacity. Besides, GMM works better on 192 filters than 128 filters, probably because 192 filters have large model capacity, so
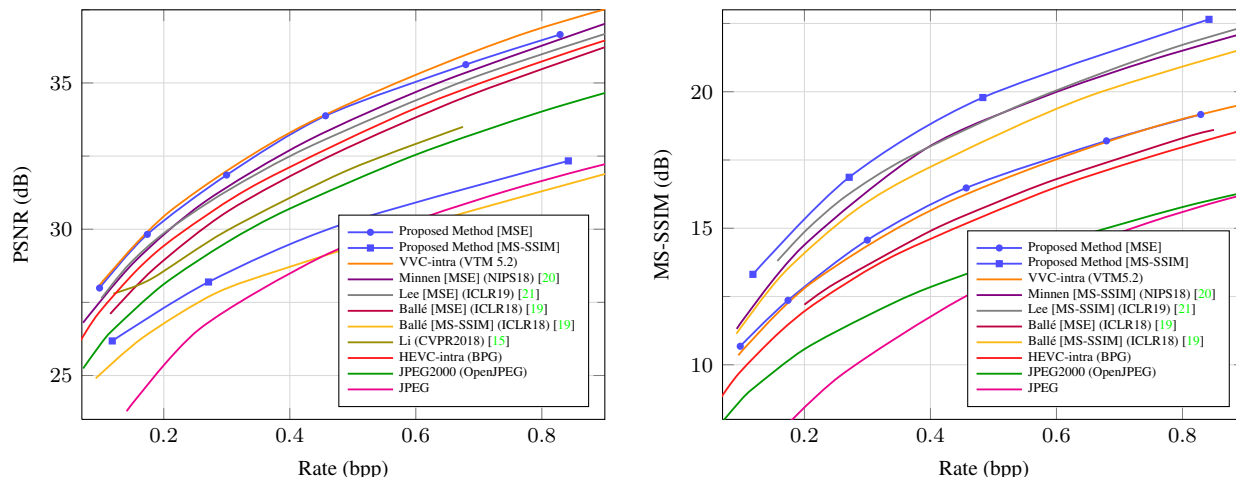
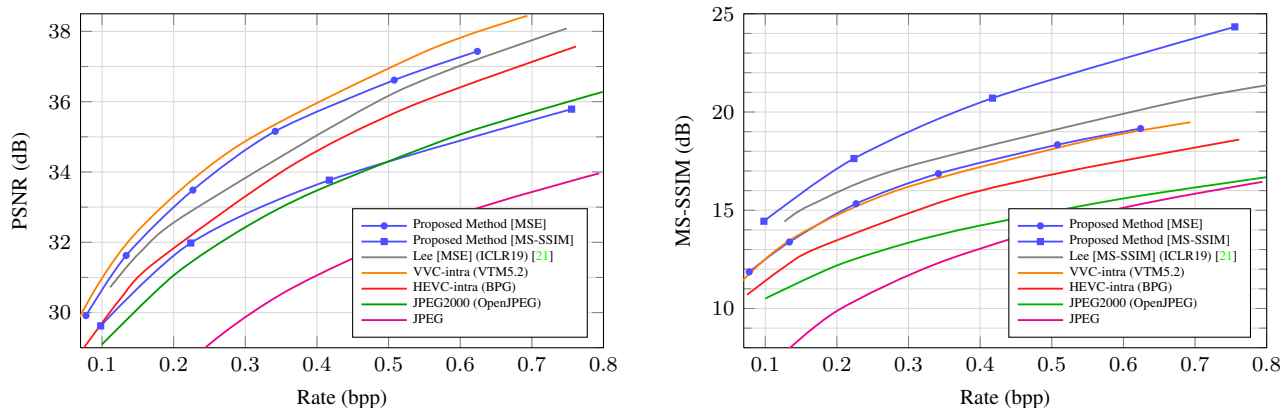Figure 7: Performance Evaluation on Kodak dataset.

Figure 8: Performance Evaluation on CLIC Professional Validation dataset.

easily resulting more remaining spatial redundancy, and our proposed Gaussian mixture likelihoods can capture them effectively to reduce the bits.

5.2. Rate-distortion Performance

The rate-distortion performance on Kodak dataset is shown in Fig. 7. MS-SSIM is converted to decibels $(-10\log_{10}(1-\text{MS-SSIM}))$ to illustrate the difference clearly. We compare our method with well-known compression standards, and recent neural network-based learned compression methods, including the works of Li *et al.* [15], Ballé *et al.* [19], Minnen *et al.* [20] and Lee *et al.* [21]. RD curves of [21] are from their released source code[1]. RD points of [19] and [20] are obtained by contacting corresponding authors. RD points of [15] are traced from their paper with only PSNR. Regarding PSNR, our method yields a competitive results with VVC and achieves better coding performance than previous learning based methods. To our

knowledge, our approach is the first work to achieve comparable PSNR with VVC. Regarding MS-SSIM, our method achieves state-of-the-art results among existing works.

Fig. 8 shows the comparison results with JPEG, JPEG2000, HEVC, VVC and the work of Lee *et al.* [21] on CLIC validation dataset. Regarding PSNR, our approach outperforms all the other codecs, except for VVC. This performance gap might be resulted from small size of training patches. Regarding MS-SSIM, our approaches significantly outperform all the codecs. It shows our method also works for high resolution images.

5.3. Qualitative Results

To demonstrate our method can generate more visually pleasant results, we visualize some reconstructed images for qualitative performance comparison.

Fig. 1 shows reconstructed images *kodim04* with approximately 0.10 bpp and a compression ratio of 240:1. More details are kept for our method optimized by MS-SSIM, and the hair appears much more natural than other codec-

---

[1] https://github.com/JooyoungLeeETRI/CA_Entropy_Model

Figure 9: Visualization of reconstructed images *kodim07* from Kodak dataset.
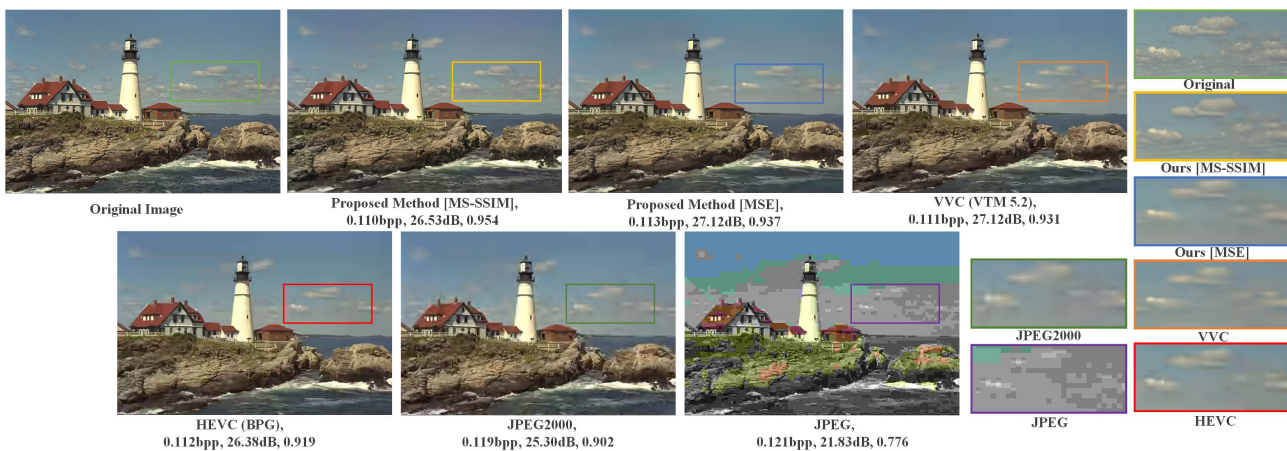

Figure 10: Visualization of reconstructed images *kodim21* from Kodak dataset.

s. Our method optimized by MSE achieves comparable results with VVC, and outperform HEVC, JPEG2000, JPEG. Fig. 9 shows reconstructed images *kodim07* with approximately 0.10 bpp and a compression ratio of 240:1. Our proposed method optimized by MS-SSIM generates more visually pleasant results, as shown by enlarged brick wall and red flowers. Our model optimized by MSE is slightly worse than VVC, but better than HEVC, JPEG2000 and JPEG. Fig. 10 shows reconstructed images *kodim21* with approximately 0.12 bpp and a compression ratio of 200:1. The shape of cloud is well preserved in our proposed method optimized by MS-SSIM. Our proposed method optimized by MSE achieves comparable quality with VVC. For the other three codecs, i.e. HEVC, JPEG2000 and JPEG, clear artifacts and blocking effect appeared. More subjective quality results are visualized in supplementary materials.

6. Conclusion

We propose a learned image compression approach using a discretized Gaussian mixture likelihoods and attention modules. By exploring the remaining redundancy of recent learned compression techniques, we have found single parameterized model can not achieve arbitrary likelihoods, limiting the accuracy of entropy models. Therefore, we use a discretized Gaussian Mixture Likelihoods to achieve a more flexible and accurate entropy model to enhance the performance. Besides, we utilize a simplified attention module with moderate complexity in our network architecture to achieve high coding efficiency.

Experimental results validate our proposed method achieves a state-of-the-art performance compared to existing learned compression methods and coding standards including HEVC, JPEG2000, JPEG. Besides, we achieve comparable performance with next-generation compression standard VVC for PSNR. The visual quality of our trained models using MS-SSIM outperforms existing methods.

Acknowledgement

References

[1] G. K Wallace, *"The JPEG still picture compression standard"*, IEEE Trans. on Consumer Electronics, vol. 38, no. 1, pp. 43-59, Feb. 1991. 1, 2

[2] Majid Rabbani, Rajan Joshi, *"An overview of the JPEG2000 still image compression standard"*, ELSEVIER Signal Processing: Image Communication, vol. 17, no, 1, pp. 3-48, Jan. 2002. 1, 2

[3] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, *"Overview of the High Efficiency Video Coding (HEVC) Standard"*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012. 1, 2

[4] G. J. Sullivan and J. R. Ohm, *"Versatile video coding Towards the next generation of video compression"*, Picture Coding Symposium, Jun. 2018. 1, 2

[5] P. Vincent, H. Larochelle, Y. Bengio and P.-A. Manzagol, *"Extracting and composing robust features with denoising autoencoders"*, Intl. conf. on Machine Learning (ICML), pp. 1096-1103, July 5-9. 2008. 2

[6] Lucas Theis, Wenzhe Shi, Andrew Cunninghan and Ferenc Huszar, *"Lossy Image Compression with Compressive Autoencoders"*, Intl. Conf. on Learning Representations (ICLR), pp. 1-19, April 24-26, 2017. 2

[7] J. Balle, Valero Laparra, Eero P. Simoncelli, *"End-to-End Optimized Image Compression"*, Intl. Conf. on Learning Representations (ICLR), pp. 1-27, April 24-26, 2017. 2

[8] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, L. V. Gool, *"Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations"*, Neural Information Processing Systems (NIPS) 2017, arXiv:1704.00648v2. 2

[9] G. Toderici, S. M.O'Malley, S. J. Hwang, et al., *"Variable rate image compression with recurrent neural networks"*, arXiv: 1511.06085, 2015. 2

[10] G, Toderici, D. Vincent, N. Johnson, et al., *"Full Resolution Image Compression with Recurrent Neural Networks"*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, July 21-26, 2017. 2

[11] Nick Johnson, Damien Vincent, David Minnen, et al., *"Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks"*, arXiv:1703.10114, pp. 1-9, March 2017. 2

[12] Ripple Oren, L. Bourdev, *"Real Time Adaptive Image Compression"*, Proc. of Machine Learning Research, Vol. 70, pp. 2922-2930, 2017. 2

[13] S. Santurkar, D. Budden, N. Shavit, *"Generative Compression"*, Picture Coding Symposium, June 24-27, 2018. 2

[14] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, *"Generative Adversarial Networks for Extreme Learned Image Compression"*, arXiv:1804.02958. 2

[15] M. Li, W. Zuo, S. Gu, D. Zhao, D. Zhang, *"Learning Convolutional Networks for Content-weighted Image Compression"*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 17-22, 2018. 2, 7

[16] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, *"Deep Convolutional AutoEncoder-based Lossy Image Compression"*, Picture Coding Symposium, pp. 1-5, June 24-27, 2018. 2

[17] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, *"Learning Image and Video Compression through Spatial-Temporal Energy Compaction"*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 16-20, 2019. 2

[18] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, *"Deep Residual Learning for Image Compression"*, CVPR Workshop, pp. 1-4, June 16-20, 2019. 2, 5

[19] Johannes Balle, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, *"Variational Image Compression with a Scale Hyperprior"*, Intl. Conf. on Learning Representations (ICLR), pp. 1-23, 2018. 1, 2, 3, 7

[20] D. Minnen, J. Ballé, G. Toderici, *"Joint Autoregressive and Hierarchical Priors for Learned Image Compression"*, arXiv.1809.02736. 1, 2, 3, 4, 7

[21] J. Lee, S. Cho, S-K Beack, *"Context-Adaptive Entropy Model for End-to-end optimized Image Compression"*, Intl. Conf. on Learning Representations (ICLR) 2019. 1, 2, 3, 7

[22] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, L. V. Gool, *"Conditional Probability Models for Deep Image Compression"*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 17-22, 2018. 2

[23] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, *"Performance Comparison of Convolutional AutoEncoders, Generative Adversarial Networks and Super-Resolution for Image Compression"*, CVPR Workshop and Challenge on Learned Image Compression (CLIC), pp. 1-4, June 17-22, 2018. 2

[24] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espehold, A. Graves, and K. Kavukcuoglu, *"Conditional Image Generation with PixelCNN Decoders"*, Advances in Neural Information Processing Systems (NIPS), 2016 2

[25] T. Salimans, A. Karpathy, X. Chen, D. P. Kingma, *"Pixel-CNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications"*, Intl. Conf. on Learning Representations (ICLR), 2017. 2, 4

[26] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, L. V. Gool, *"Practical Full Resolution Learned Lossless Image Compression"*, CVPR 2019. 2

[27] Goyal Vivek K. *"Theoretical Foundations of Transform Coding"*, IEEE Signal Processing Magazine, Vol. 18, No. 5, Sep. 2001. 2

[28] Rissanen Jorma and Glen G. Langdon Jr., *"Universal modeling and coding"*, IEEE Transactions on Information Theory, Vol. 27, No. 1, Jan. 1981. 3

[29] Y.Zhang, K. Li, K. Li, B. Zhong, Y. Fu, *"Residual Non-local Attention Networks for Image Restoration"*, Intl. Conf. on Learning Representations (ICLR), pp. 1-18, 2019. 5

[30] H. Liu, T. Chen, P. Guo, Q. Shen, X. Cao, Y. Wang, Z. Ma, *"Non-local Attention Optimized Deep Image Compression"*, arXiv.1904.09757. 5

[31] J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei, *"ImageNet: A Large-Scale Hierarchical Image Database"*, IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1-8, June 20-25, 2009. 6

[32] D. P. Kingma and J. Ba, *"Adam: A method for stochastic optimization"*, arXiv:1412.6980, pp.1-15, Dec. 2014. 6

[33] Z. Wang, E. P. Simoncelli and A. C. Bovik, *"Multiscale structural similarity for image quality assessment"*, The 36-th Asilomar Conference on Signals, Systems and Computers, Vol.2, pp. 1398-1402, Nov. 2013. 6

[34] Kodak Lossless True Color Image Suite, Download from http://r0k.us/graphics/kodak/ 6

[35] Workshop and Challenge on Learned Image Compression, CVPR2018, http://www.compression.cc/challenge/ 6

[36] VVC Official Test Model VTM, https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tree/VTM-5.2, accessed on July, 2019. 6

[37] BPG Image Format, https://bellard.org/bpg/ 6

[38] JPEG2000 official software OpenJPEG, https://jpeg.org/jpeg2000/software.html 6

[39] Python Imaging Library (PIL), https://pillow.readthedocs.io/en/5.1.x/index.html 6