Received 10 June 2023, accepted 26 June 2023, date of publication 3 July 2023, date of current version 21 July 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3291591

RESEARCH ARTICLE

Learned Lossless Image Compression With Combined Channel-Conditioning Models and Autoregressive Modules

RAN WANG[1], JINMING LIU[2], HEMING SUN[3], (Member, IEEE), AND JIRO KATTO[1,4], (Member, IEEE)
[1]Department of Computer Science and Communication Engineering, Waseda University, Tokyo 169-8050, Japan
[2]Shanghai Jiao Tong University, Shanghai 200240, China
[3]Faculty of Engineering, Yokohama National University, Yokohama 240-8501, Japan
[4]Waseda Research Institute for Science and Engineering, Waseda University, Tokyo 169-8050, Japan
Corresponding author: Ran Wang (wangran.candice@akane.waseda.jp)

This work was supported by NICT, Japan, under Grant 03801.

ABSTRACT Lossless image compression is an important research field in image compression. Recently, learning-based lossless image compression methods achieved impressive performance compared with traditional lossless methods, such as WebP, JPEG2000, and FLIF. The aim of the lossless image compression algorithms is to use shorter codelength to represent images. To encode an image with fewer bytes, eliminating the redundancies among the pixels in the image is highly important. Hence, in this paper, we explore the idea of combining an autoregressive model for the raw images based on the end-to-end lossless architecture proposed to enhance the performance. Furthermore, inspired by the successful achievements of Channel-conditioning models, we propose a Multivariant Mixture distribution Channel-conditioning model (MMCC) in our network architecture to boost performance. The experimental results show that our approach outperforms most classical lossless compression methods and existing learning-based lossless methods.

INDEX TERMS Lossless image compression, autoregressive model, channel-conditioning model.

I. INTRODUCTION

Image compression is an important task in many research fields. Along with the development of technology, more and more fields have greater demand for image compression. In many important technical fields, such as medicine, remote sensing, details of the image are crucial. Compared with lossy image compression, lossless image compression can preserve all details in the picture. Therefore, lossless compression is a crucial research topic in the above fields. As well as lossy image compression, lossless compression tries to capture the spatial correlations of the image to reduce the spatial redundancies in the compressed bitstream. Therefore, how to design the architecture of the network to reduce redundancy is the main task of image compression. JPEG2000 [1], WebP [2], and FLIF [3] are representative traditional compression methods. They rely on hand-crafted designed encoder and decoder to capture the spatial correlation between pixels. For instance, FLIF is the current state-of-the-art non-learned algorithm. It relies on a well-designed entropy coding method called "meta-adaptive near-zero integer arithmetic coding" (MANIAC). MANIAC is a dynamic data structure utilized as a context model in FLIF. With MANIAC, FLIF achieves remarkable performance.

With the development of deep learning in recent years, learning-based image compression methods have achieved a better performance than classical compression methods. The critical aim of both traditional and learning-based lossless image compression methods is to find an appropriate distribution that is as close to the real distribution as possible. Flow-based models admit exact likelihood optimization with bijective mappings. iVPF [22] is one of the remarkable flow-based methods. In [22], Zhang et al. proposed the Modular Affine Transformation (MAT) algorithm, which achieves exact bijective mapping without any numerical

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

error. As for other learning-based lossless compression methods, [10], [11] also obtain an impressive performance. L3C [10] that can propose a fully parallel hierarchical probabilistic model can outperform WebP, JPEG2000, and PNG. In [11], RC leverages BPG to obtain a lossy reconstruction and uses the proposed RC (Residual Compressor) network to achieve lossless compression. In [12], it proposed an end-to-end lossless image compression framework based on their lossy image compression work [9]. Reference [12] utilized an autoregressive model for the latents to lift the performance. Utilizing the autoregressive model could help the model to obtain an accurate probability by estimating the discrete probability of the raw pixel values. The probability of pixels in image is modelled in sequence. Although autoregressive models are powerful methods, they are very time-consuming.

In this paper, we consider data compression with channel-wise architecture and widely used architecture in lossy image compression, such as [8], and [9], into our framework. First, we utilize channel-conditional (CC) models [5] to replace the autoregressive context model for the latents to solve the time-consuming disadvantage of utilizing the autoregressive model. CC is proposed to solve the shortcoming that autoregressive models decode each symbol in sequence. As well as autoregressive models, CC models capture the redundancies between pixels. Therefore, CC models could be interpreted as efficient autoregressive models along the channel dimension. Second, original channel-conditioning models adopt the SGM-based entropy model, the performance of SGM distribution is not as good as multivariant mixture distribution, such as the Gaussian mixture model (GMM). Therefore, in this paper, we propose a multivariant mixture distributions CC model (MMCC) to better approximate the real distribution.

We test our method on DIV2K [16], CLICP, and CLICM [17]. The result shows that our method outperforms most traditional lossless image compression methods, such as PNG [6], JPEG 2000, and FLIF [3]. For learning-based methods, our method achieved better performance compared with L3C [10] and RC [11].

II. RELATED WORKS

A. LOSSLESS COMPRESSION ALGORITHMS

According to Shannon's source coding theory [23], the codelength of lossless image compression could be formulated as:

$$H\left(p\right)=\mathbb{E}_{p(x)}[-log_2 p(x)] \tag{1}$$

$p(x)$ is the distribution of raw images x. The ideal situation is that the distribution we choose is the real distribution of raw images. However, the real distribution of raw images is intractable. To use shorter codelength to encode raw images, it is crucial that the distribution we choose is accurate. Generative models are often used to estimate the probability of input data in lossless image compression. In addition, there are several other methods that have been successfully applied to lossless image compression.

1) LOSSLESS CODING

The aim of data/image compression is that using fewer bits to represents original data or images. The advantages of data/image compression are the reduced data transmission time and communication bandwidth. There are several approaches to lossless code images. Traditional entropy coding methods include Hoffman coding [7], AC (Arithmetic Coding) [24], and asymmetric numeral system (ANS) [25]. [31] proposes a lossless data compression method using machine learning. The model used is a sequence-to-sequence recurrent neural network (RNN) model for both compression and decompression.

2) GENERATIVE METHODS

Flow-based lossless compression methods like IDF [21], and iVPF [22] achieve impressive performance. Lossless compression requires discrete data for entropy coding, while common flows are continuous. To address this issue, IDF [21] proposed an invertible mapping between discrete data and latent variables. iVPF [22] explored the effectiveness of volume-preserving flow for lossless compression with the proposed novel computation algorithm Modular Affine Transformation (MAT). Experiments show that iVPF achieves impressive performance among flow-based lossless algorithms. However, most flow-based lossless compression methods have some constraints on flow layers to ensure bijections between input data and the latents, which limits the feasibility of the performance.

3) AUTOREGRESSIVE METHODS

The autoregressive model is not only widely used in image compression but also in image generation and super-resolution. Gated PixelCNN [18], PixcelCNN [19], and PixelCNN++ [20] are impressive works of the autoregressive model. PixelCNN [19] estimates the joint distribution $p(x)$ of the current pixel conditions on all the previously generated pixels left and above the current pixel:

$$p\left(x\right)=\prod_{i=1}^{n^2} p(x_i|x_1,\cdots,x_{i-1}) \tag{2}$$

The value $p(x_i|x_1,\cdots,x_{i-1})$ is the probability of the $i-th$ pixel x_i given all the previous pixels $x_1,\ldots,x_{i-1}$. The aim of utilizing autoregressive models is to capture the spatial correlations between pixels and eliminate the redundancies among the latents. With the help of autoregressive methods, we could provide a more accurate distribution for the encoder and decoder to compress images with less bits.

L3C [10] can be deemed as a VAE-based hierarchical probabilistic model with auxiliary feature representations. The proposed struct in [12] can be regarded as an optimized VAE-based end-to-end framework for the lossless image compression task.

The framework of GOLLIC [29] utilizes a three-level hierarchical framework based on the L3C [10]. A self-supervised

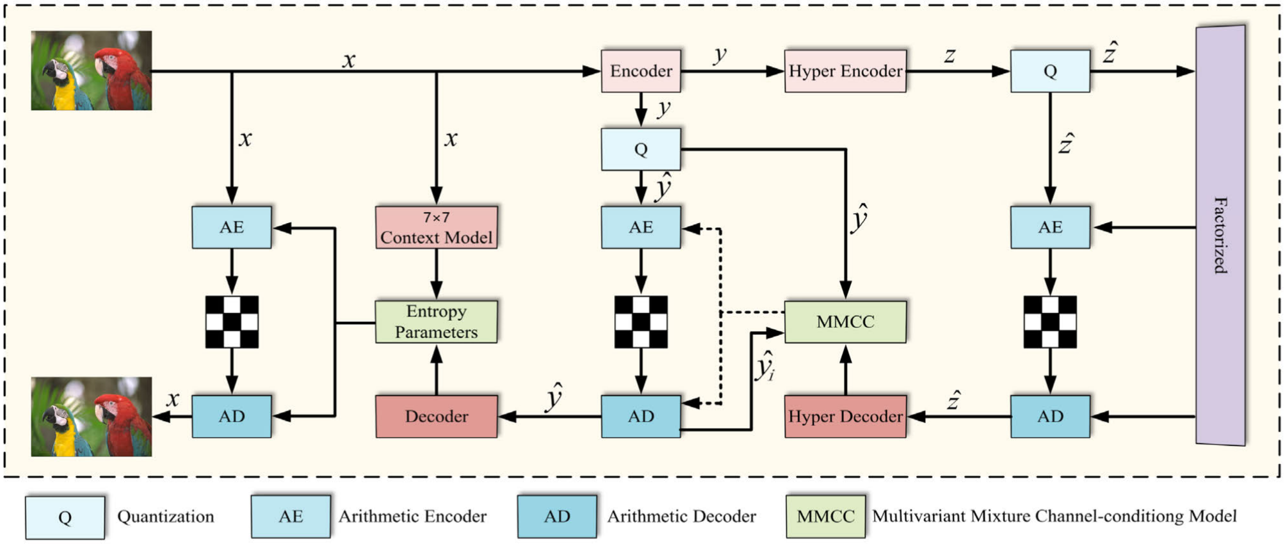


FIGURE 1. Network architecture.

clustering module is introduced in GOLLIC to capture the long-term dependencies inside the image.

4) OTHER METHODS

RC [11] leverages BPG to compress raw images into lossy images and calculates the residuals between the raw images and lossy images. The residuals are then compressed with a convolutional neural network. Recent works also introduce classical machine learning methods like K-Mean clustering to boost performance [29], [32]. Reference [33] employs a deep learning-based approach which built on the ML concepts of RESL framework for computing the residual-error for a dual prediction method like RC. Reference [34] proposed a lossy architecture plus residual coding which could achieve both lossless and near-lossless image compression. Reference [35] proposed a lossless image compression framework that decomposes an input image into low-frequency and high-frequency regions, enabling a coarse-to-fine processing approach. Reference [36] developed an efficient end-to-end generative model-based architecture for lossless image compression.

B. CONTEXT MODEL

The context model is used to predict the probability of unknown codes based on latents that have already been decoded. This method is boost in [8], where hyper latent and context are used jointly to predict both the mean and scale parameter of the entropy model. The adopted autoregressive context model in [8] explores the spatial dimension. In their following work, Channel-conditioning model (CC model) is proposed in [5]. CC model could be regarded as a different type of autoregressive model, in which explores the correlations among pixels along the channel dimension rather than the spatial dimensions. CC model first divides channels into N slices. Then CC model compresses the first slide solely based on the information provided by the hyperprior. And the hyperprior and decompressed first slide are utilized to encoder and decoder the second slice, and so on. Finally, all the decoded slices are concatenated to form the final output and transformed into the final reconstructed image.

Inspired by the impressive performance achieved by the CC model in lossy image compression. We proposed a multivariant mixture distribution CC model (MMCC) to boost the performance of our lossless image compression work. The details will be discussed in the Method section.

C. FORMULATION OF LEARNING-BASED IMAGE COMPRESSION

Our framework is inspired by the framework widely adopted in learning-based lossy compression methods [8], [9]. In the framework of a classical learning-based lossy image compression, the operation can be formulated as [13]:

$$\begin{aligned} y &= g_a(x;\phi) \\ \hat{y} &= Q(y) \\ \hat{x} &= g_s(\hat{y};\theta) \end{aligned} \tag{3}$$

where x stands for the raw images, and $\hat{x}$ is the reconstructed images. The latents presentation before quantization and compressed codes are denoted as y and $\hat{y}$, respectively. ϕ and θ denote the parameters that need to be optimized for analysis and synthesis transforms. Q denotes the quantization, and $U|Q$ represents quantization and entropy coding. The input image x is first encoded as the latent representations y through an analysis transform $g_a(x;\phi)$. Then the latents y are quantized to discrete values $\hat{y}$, and $\hat{y}$ will be losslessly coded with the hyperprior. On the decoder side, the reconstructed image $\hat{x}$ can be recovered with the synthesis transform $g_s(\hat{y};\theta)$. In [27], a hyperprior entropy model is proposed by adding a hierarchical autoencoder. In addition, a conditional Gaussian scale mixture model with zero-mean and scale parameters σ^2 was also introduced to the model

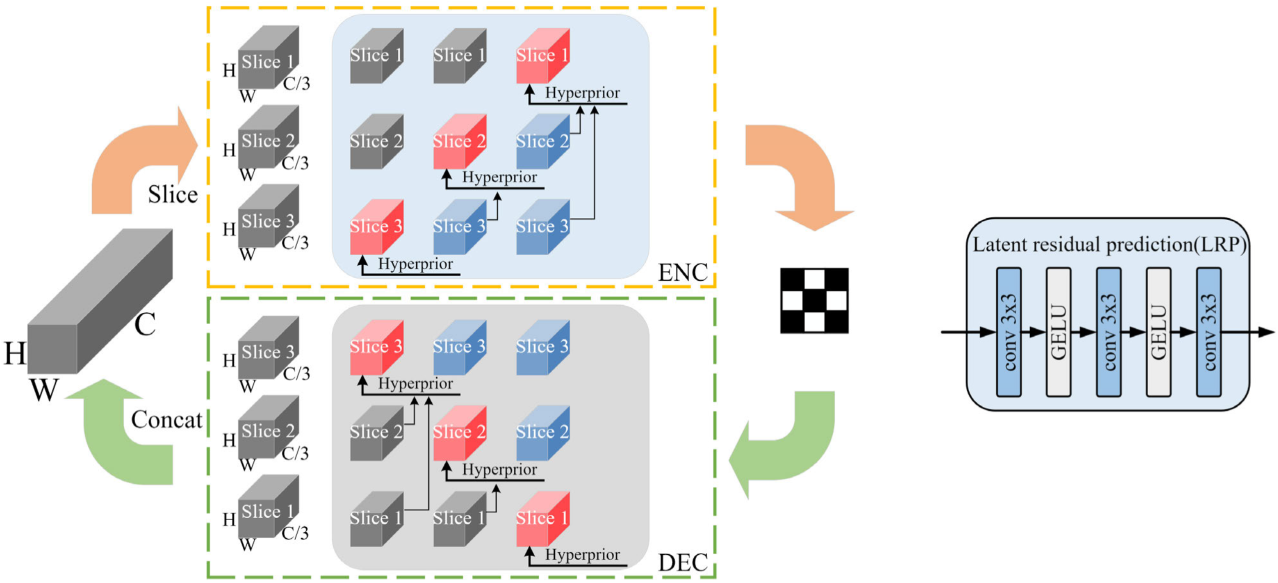


FIGURE 2. (a) The Workflow of MMCC. ENC stands for encoder block and DEC represents decoder block. (b) The architecture of LRP (latent residual prediction).

$\hat{y}$ in [27]. Based on [27], an enhanced entropy model with an autoregressive context model is proposed in [8]. Following that, [9] proposed a learning-based image compression with a discretized Gaussian mixture likelihoods and attention modules.

Unlike lossy image compression, in lossless image compression, errors are not acceptable. Therefore, in our framework, instead of outputting reconstructed images, the output $\hat{y}$ is the predicted probability for raw images x. Since we have the probability of x, we could lossless encode the images x. The structure we proposed can be deemed as a hierarchical VAE-based compression method.

III. METHOD

In this section, we first introduce the details of proposed MMCC model and autoregressive model adopted for raw image. Finally, we introduce the L2-norm that we utilized to speed up training.

A. FORMULATION OF LOSSLESS IMAGE COMPRESSION

The workflow in this study follows a similar approach to that of lossy image compression, as illustrated in Figure 1. The lossless image compression consists of two components: the main path and the hyper path. The main path can be described as follows:

$$\begin{aligned} y &= g_a(x; \Phi) \\ \hat{y} &= Q(y) \\ (\mu_x, \sigma_x, \pi_x) &= g_s(\hat{y}; \theta) \end{aligned} \tag{4}$$

where the variables x, y, and $\hat{y}$ represent a raw image, a latent presentation before quantization, and a quantized latent presentation, respectively. ϕ and θ are trainable parameters of the encoder g_a and decoder g_s.

The latent representation, denoted as y, is generated by feeding the raw images x into the encoder g_a. To encode the latent representation y, it needs to undergo a quantization process denoted by Q. The output of the quantization operator Q is represented as $\hat{y}$. To achieve lossless encoding and decoding of $\hat{y}$, we model its distribution as a Gaussian mixture distribution with parameters μ_y, σ_y, and π_y which are generated by the proposed Multivariate Mixture CC Model (MMCC). Subsequently, $\hat{y}$ is transmitted to the decoder g_s. Unlike lossy image compression, the output of the decoder g_s consists of parameters that are used for lossless encoding of the raw images.

The hyperprior path is consisted of the hyper encoder h_a and the hyper decoder h_s. It can be formulated as follows:

$$\begin{aligned} z &= h_a(y; \Phi_h) \\ \hat{z} &= Q(z) \\ (\tilde{\mu}_y, \tilde{\sigma}_y, \tilde{\pi}_y) &= h_s(\hat{z}; \theta_h) \end{aligned} \tag{5}$$

where z and $\hat{z}$ represent a hyperprior presentation before quantization and a quantized hyperprior presentation, respectively. The parameters $\tilde{\mu}_y$, $\tilde{\sigma}_y$, and $\tilde{\pi}_y$ are generated by the hyper decoder and will later be used as inputs for the proposed Multivariate Mixture CC Model (MMCC) along with the sliced y. The detailed explanation of MMCC will be provided in the next section.

To modify the quantization errors $(y - \hat{y})$, we employ the Latent Residual Prediction (LRP) method, which is further elaborated in Figure 2.

B. MULTIVARIATE MIXTURE CC MODEL

The context model plays a crucial role in estimating the feature parameters. In Figure 2, we present the workflow of the Multivariate Mixture CC Model (MMCC). The input to

MMCC consists of the parameters generated by the hyper decoder and the quantized latents y. The latents y are evenly split into N slices along the channel dimensions, with each slice containing $W \times H \times C/N$ values.

The encoding and decoding process of the slices in MMCC follows a sequential dependency. The first slice, y_0, is encoded and decoded solely based on the hyperprior. The second slice, y_1, is encoded and decoded considering both the first slice and the hyperprior, and so on. This process can be formulated as:

$$\begin{aligned} y &= \{y_0, y_1, \ldots, y_{N-1}\} \\ \left(\mu_{y_i}, \sigma_{y_i}, \pi_{y_i}\right) &= MMCC\left(\tilde{\mu}_y, \tilde{\sigma}_y, \tilde{\pi}_y, \tilde{y}_{<i}\right) \\ \tilde{y}_i &= LRP(y_i) \\ \tilde{y} &= \{\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_{N-1}\} \\ \mu_y &= concat(\mu_{y_i}) \\ \sigma_y &= concat(\sigma_{y_i}) \\ \pi_y &= concat(\pi_{y_i}) \end{aligned} \tag{6}$$

where y represents the latents, $y_0, y_1, \ldots, y_{N-1}$ denote the individual slices of y, $\tilde{y}_i$ denotes the output of the LRP, which signifies the variable y_i modified by incorporating the quantization error using the LRP method. Additionally, $\tilde{y}_{<i}$ represents the slices before the $i-th$ index that have been modified by the quantization error. μ_{y_i}, σ_{y_i}, and π_{y_i} denote the slice of μ_y, σ_y, and π_y, respectively. μ_y, σ_y, and π_y denote the parameters of the Gaussian mixture distribution utilized for the encoding and decoding of the latents y. This process shares similarities with the autoregressive context model, although the autoregressive model operates among spatial dimensions. The original CC model, proposed earlier, used the Signal Gaussian distribution for the hyperprior model. However, previous research has demonstrated the effectiveness of multivariate mixture distributions [12]. Therefore, in our work, we introduce the Multivariate Mixture CC Model (MMCC) to further enhance the performance of our model.

The motivation behind considering a Gaussian Mixture Model (GMM) is to create a more flexible parameterized model capable of achieving arbitrary likelihoods. GMMs can provide improved accuracy in complex areas such as boundaries and edges, which makes them well-suited for our purposes.

C. AUTOREGRESSIVE CONTEXT MODEL FOR RAW IMAGE

Autoregressive context model can effectively capture spatial correlations to predict current pixel, which is like classical intra prediction. Autoregressive context model is usually implemented in the format of a $N \times N$ mask convolution. More details about mask convolution could be found in [8]. Different with the latents y, the input image x only has three channels. The redundancies mostly exist in spatial dimensions. We tried to increase the number of channels of x by imply space-to-depth operation [21]. However, the performance becomes worse, we speculate the reason is that space-to-depth operation destroy the structure of the original image and the model could not learn an accurate distribution. Therefore, to eliminate the redundancies in raw image, we adopt an 7×7 autoregressive context model to x.

D. LOSS FUNCTION

Different lossy image compression, if we use the lossy loss function to train our lossless image compression model. It takes quite a long time to find a proper and accurate distribution during the training. To speed up this process, we leverage the L2-norm. L2-norm calculates the difference between the ground-truth value and the predicted mean value, the formulation is shown in the following:

$$L' = \parallel \mu_x - x \parallel^2 + \parallel \mu_y - \hat{y} \parallel^2 \tag{7}$$

Hence, the loss function for training could be formulated as:

$$\begin{aligned} L = &\mathbb{E}\left[-log_2\left(p_x\left(x\,|\,\hat{y}\right)\right)\right] + \mathbb{E}\left[-log_2\left(p_{\hat{y}}\left(\hat{y}\,|\,\hat{z}\right)\right)\right] \\ &+ \mathbb{E}\left[-log_2\left(p_{\hat{z}}\left(\hat{z}\,|\,\hat{\varphi}\right)\right)\right] + \lambda \cdot (\parallel \mu_x - x \parallel^2 + \parallel \mu_y - \hat{y} \parallel^2) \end{aligned} \tag{8}$$

λ responses to control the weights of L2-norm term. Normally, λ sets as 0.6 for the first 20 epochs. After 20 epochs, we choose to set λ as 0. This L2-norm shares the same idea with mean square error (MSE) loss in lossy image compression. The details of L2-norm are discussed in [12].

IV. EXPERIMENTS

A. TRAINING DETAILS

Our model is built on the architecture of Cheng2020 in CompressAI platform [30]. It is worth noting that the output channel number of g_s is $3 \times 3 \times K$, where K represents the parameter chosen for GMM. For the purposes of this paper, the value of K for GMM has been set to 3. For training, we choose about 40000 images from the ImageNet [14] and cropped the size to 256×256 before randomly feeding them into the network. The total number of parameters is 709M in our model. These images are not ideal for the lossless compression task, but we are not aware of a similarly large-scale lossless training data set. We set the number of channels $N = 192$ for main path and $M = 320$ for hyper path. The number of slices is 10. During the training, the number of K for GMM is set as 3.

The model was optimized using Adam [15] with a batch size of 8, and the learning rate was set to 1×10^{-4} at the beginning of training. After 100 epochs, the learning rate dropped to 1×10^{-5}. Our model was trained about 400 epochs to obtain stable performance. We ran all the experiments by the machine equipped with an NVIDIA GeForce RTX 3090 with 24GBs of memory.

B. EVALUATION

For comparison, we evaluate our model on three widely used lossless high-resolution image compression datasets, CLIC.pro (CLIC.P), CLIC.mobile (CLIC.M), DIV2K.

TABLE 1. Compression performance on different datasets.

Model	CLICP (bpsp)	CLICM (bpsp)	DIV2K (bpsp)	CIFAR10 (bpsp)	Kodak (bpsp)
Ours	**2.63**	**2.43**	**2.74**	**4.89**	**2.94**
Cheng [12]	3.36	3.23	3.48	-	-
RC [10]	2.93	2.54	3.08	-	-
L3C [11]	2.94	2.64	3.09	-	3.26
iVPF [22]	2.39	2.54	2.68	3.20	-
GOLLIC [29]	2.83	2.62	3.07	-	-
FLIF [3]	2.78	2.49	2.91	4.19	2.90
PNG [4]	4.00	3.90	4.24	5.87	4.35

In addition, we also evaluate our model on Kodak [6] and CIFAR10. All the results are measured by bits per sub-pixel (bpsp) in our paper.

CLIC.mobile and CLIC.pro are released as part of the "Workshop and Challenge on Learned Image Compression" (CLIC). CLIC.mobile contains 61 images taken using cell phones, while CLIC.pro contains 41 images from DSLRs, retouched by professionals. DIV2K is a super-resolution dataset with high-quality images. Kodak dataset [6] consists of 24 uncompressed 768×512 color images, widely used in evaluating lossy image compression methods. The CIFAR10 is a low-resolution dataset consists of 60000 32×32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The models shown in the Table 1, we only report the compression performance published by their authors, because their codes are either difficult to be generalized to arbitrary datasets or unavailable.

It can be observed from Table 1 that our method outperforms the classical methods PNG [4] for all datasets. FLIF is the state-of-art classical lossless image compression method. Our proposed model lift about 5.4% performance compared with FLIF among the three high-resolution datasets. As for Kodak and CIFAR10, FILF outperform our model slightly.

For the learning-based methods, we lift about 30% and 12% performance compared with Cheng et al. [12] and L3C [10] in high-resolution datasets, respectively. GOLLIC [29] and RC [11] are both an enhanced work based on L3C [10], the results show that our work outperforms RC and GOLLIC for all three high-resolution datasets. As for the state-of-the-art flow-based model iVPF, it outperforms our model. For high-resolution datasets, our model is slightly worse compared with iVPF. As for CIFAR10 dataset, the performance of our model is not ideal. It's worth noting that iVPF is trained on low-resolution datasets with 64×64 patch size while our model is trained on 256×256 patch size. Since iVPF has not open source, we are unable to compare the inference time with it. In Table 5, we compare inference time with another flow-based model IDF [21]. Inference time measures evaluation of 10,000 test images with a batch size of 100. It can be observed that our model has less inference time.

V. ABLATION STUDY

A. AUTOREGRESSIVE CONTEXT MODEL

From Table 2, it can be observed that the model with an extra autoregressive model for x lifts the performance of compression.

TABLE 2. Performance of different models.

Model	CLIC.P (bpsp)	CLIC.M (bpsp)	DIV2K (bpsp)
[12]	3.282	3.134	3.392
[12] + 7×7 autoregressive model for x	2.839	2.632	2.942
MMCC + 7×7 autoregressive model for x (ours)	**2.621**	**2.414**	**2.729**

TABLE 3. Performance of SGM-based CC models and MMCC models.

Models	CLIC.P (bpsp)	CLIC.M (bpsp)	DIV2K (bpsp)
SGM-based CC	3.015	2.824	3.125
MMCC (K=3)	**2.621**	**2.414**	**2.729**
MMCC (K=5)	2.721	2.528	2.822

In Table 4, we shown the performance with different size of autoregressive context model. The baseline in Table 4 is [12]. Compared with 5×5 masked convolution, 7×7 masked convolution achieves better performance. We only explore the size of 5×5 and 7×7 masked convolution which are normally adopted size of masked convolution.

B. MMCC MODEL

GMM is sensitive to the selection of the value of K. In Table 3, we explore the different setting number K of GMM. In the

TABLE 4. Bits allocation for different parts.

Models	CLICP (bpsp)	CLICM (bpsp)	DIV2K (bpsp)
[12] + 5×5 autoregressive model for x	2.942	2.759	3.064
[12] + **7×7** autoregressive model for x	2.839	2.632	2.942

TABLE 5. Inference time on CIFAR10.

Models	Our	IDF
Time (s)	0.23	20.58

first row, SGM-based CC means the value of K is 1. From the table that we could observed that the best performance is when K equal to 3.

VI. CONCLUSION

In this paper, we explore the idea of adding an autoregressive model for the raw images to reduce the redundancies between pixels. Furthermore, inspired by the impressive performance the CC model achieved. We proposed the MMCC model for the latents. Experiments demonstrate that our proposed method outperforms classical lossless image compression methods, such as PNG and FLIF. For learning-based methods, we outperform L3C and RC for all test datasets. Furthermore, the image size of evaluation datasets for lossless image compression is much larger compared with lossy image compression. Hence, to further improve our work, we could investigate different forms of context models to capture global-scope spatial correlations and cross-channel relationships between the pixels. Moreover, it would be interesting to explore domain-specific applications, e.g., for medical image data and remote sensing.

REFERENCES

[1] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *Signal Process., Image Commun.*, vol. 17, no. 1, pp. 3–48, Jan. 2002. [Online]. Available: https://jpeg.org/jpeg2000/software.html

[2] *WebP Image Format*. Accessed: Jul. 16, 2023. [Online]. Available: https://developers.google.com/speed/webp/

[3] J. Sneyers and P. Wuille, "FLIF: Free lossless image format based on MANIAC compression," in *Proc. ICIP*, 2016, pp. 66–70. [Online]. Available: https://flif.info/

[4] *Portable Network Graphics (PNG)*. Accessed: Jul. 16, 2023. [Online]. Available: http://libpng.org/pub/png/libpng.html

[5] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3339–3343.

[6] E. Kodak. (1993). *Kodak Lossless True Color Image Suite (PHOTOCD PCD0992)*. [Online]. Available: http://r0k.us/graphics/kodak/

[7] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1102, Sep. 1952.

[8] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. NIPS*, 2018, pp. 1–10.

[9] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7936–7945.

[10] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Practical full resolution learned lossless image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10621–10630.

[11] F. Mentzer, L. Van Gool, and M. Tschannen, "Learning better lossless compression using lossy compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6637–6646.

[12] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned lossless image compression with a hyperprior and discretized Gaussian mixture likelihoods," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2158–2162.

[13] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 9–21, Sep. 2001.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1–9.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[16] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131.

[17] *Workshop and Challenge on Learned Image Compression*. Accessed: Jul. 16, 2023. [Online]. Available: https://www.compression.cc/

[18] A. V. D. Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with PixelCNN decoders," in *Proc. NIPS*, 2016, pp. 1–9.

[19] A. Kolesnikov and C. H. Lampert, "PixelCNN models with auxiliary variables for natural image modeling," in *Proc. ICML*, 2017, pp. 1905–1914.

[20] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications," in *Proc. ICLR*, 2017, pp. 1–10.

[21] E. Hoogeboom, J. Peters, R. Van Den Berg, and M. Welling, "Integer discrete flows and lossless compression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[22] S. Zhang, C. Zhang, N. Kang, and Z. Li, "IVPF: Numerical invertible volume preserving flow for efficient lossless compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1–10.

[23] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[24] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[25] J. Duda, "Asymmetric numeral systems," 2009, *arXiv:0902.0271*.

[26] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14771–14780.

[27] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. ICLR*, 2018, pp. 1–23.

[28] *Kodak Lossless True Color Image Suite*. Accessed: Jul. 16, 2023. [Online]. Available: http://r0k.us/graphics/kodak/

[29] Y. Lan, L. Qin, Z. Sun, Y. Xiang, and J. Sun, "GOLLIC: Learning global context beyond patches for lossless high-resolution image compression," 2022, *arXiv:2210.03301*.

[30] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "CompressAI: A PyTorch library and evaluation platform for end-to-end compression research," 2020, *arXiv:2011.03029*.

[31] I. Schiopu and A. Munteanu, "Deep-learning-based lossless image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 1829–1842, Jul. 2020.

[32] E. V. S. Rochishna, V. V. N. S. M. P. G. H. P. Rao, G. Bhargav, and S. Sathyalakshmi, Eds., *Lossless Image Compression Using Machine Learning*. Singapore: Springer, 2023. Accessed: Mar. 24, 2023, doi: 10.1007/978-981-19-5443-6_10.

[33] R. Barman, S. Badade, S. Deshpande, S. Agarwal, and N. Kulkarni, Eds., *Lossless Data Compression Method Using Deep Learning*. Singapore: Springer, 2022. Accessed: Mar. 24, 2023, doi: 10.1007/978-981-16-9650-3_11.

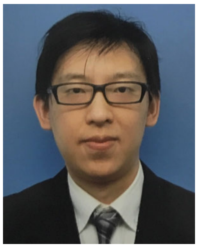
[34] Y. Bai, X. Liu, K. Wang, X. Ji, X. Wu, and W. Gao, "Deep lossy plus residual coding for lossless and near-lossless image compression," 2022, *arXiv:2209.04847*.

[35] H. Rhee, Y. I. Jang, S. Kim, and N. I. Cho, "LC-FDNet: Learned lossless image compression with frequency decomposition network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6033–6042.

[36] N. Kang, S. Qiu, S. Zhang, Z. Li, and S. Xia, "PILC: Practical image lossless compression with an end-to-end GPU oriented neural framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3739–3748.

RAN WANG received the B.S. degree from the Ocean University of China and the master's degree from the University of Chinese Academy of Sciences (UCAS). She is currently pursuing the Ph.D. degree with Waseda University. Her current research interest includes lossless image compression.

JINMING LIU received the B.S. degree from Southeast University and the M.Sc. degree from Waseda University. He is currently pursuing the Ph.D. degree with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His current research interests include image/video compression, computer vision, and deep learning.



HEMING SUN (Member, IEEE) received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011, and the dual M.E. degree from Waseda University and Shanghai Jiao Tong University, in 2012 and 2014, respectively, and the Ph.D. degree in embodiment informatics from Waseda University, in 2017. He was a Researcher with the NEC Central Research Laboratories, from 2017 to 2018. He is selected as the Japan Science and Technology Agency (JST) PRESTO Researcher, from 2019 to 2023. He is currently an Assistant Professor with Waseda University. His current research interests include algorithms and VLSI architectures for image/video processing and neural networks. He got several awards, including the IEEE Computer Society Japan Chapter Young Author Award, the VCIP Best Paper Award, the VCIP Best Reviewer Award, the PCS Best Paper Finalist, and the ISSCC Takuo Sugano Award for Outstanding Far-East Paper.

JIRO KATTO (Member, IEEE) received the B.S., M.E., and Ph.D. degrees from The University of Tokyo, in 1987, 1989, and 1992, respectively. He was with NEC Corporation, from 1992 to 1999. He was a Visiting Scholar with Princeton University, NJ, USA, from 1996 to 1997. He joined as an Associate Professor with Waseda University, in 1999, where he has been a Professor, since 2004. He was the Director of the New Energy and Industrial Technology Development Organization (NEDO), from 2004 to 2008. He has been a Program Officer of the Japan Society for the Promotion of Science (JSPS), since 2019. His current research interests include multimedia signal processing and multimedia communication systems. He is a member of the Association for Computing Machinery (ACM) and the Information Processing Society of Japan (IPSJ). He is a fellow of the Institute of Electronics, Information and Communication Engineers (IEICE) and the Institute of Image Information and Television Engineers (ITE). He was the AHG Chair/Co-Chair of ISO/IEC JTC1 SC29/WG11(MPEG), from 1996 to 1997.