Fast and High-Performance Learned Image Compression With Improved Checkerboard Context Model, Deformable Residual Module, and Knowledge Distillation

Haisheng Fu, Feng Liang, Jie Liang, Yongqiang Wang, Zhenman Fang, Guohe Zhang, Jingning Han

***Abstract*—Deep learning-based image compression has made great progresses recently. However, some leading schemes use serial context-adaptive entropy model to improve the rate-distortion (R-D) performance, which is very slow. In addition, the complexities of the encoding and decoding networks are quite high and not suitable for many practical applications. In this paper, we propose four techniques to balance the trade-off between the complexity and performance. We first introduce the deformable residual module to remove more redundancies in the input image, thereby enhancing compression performance. Second, we design an improved checkerboard context model with two separate distribution parameter estimation networks and different probability models, which enables parallel decoding without sacrificing the performance compared to the sequential context-adaptive model. Third, we develop a three-pass knowledge distillation scheme to retrain the decoder and entropy coding, and reduce the complexity of the core decoder network, which transfers both the final and intermediate results of the teacher network to the student network to improve its performance. Fourth, we introduce L_1 regularization to make the numerical values of the latent representation more sparse, and we only encode non-zero channels in the encoding and decoding process to reduce the bit rate. This also reduces the encoding and decoding time. Experiments show that compared to the state-of-the-art learned image coding scheme, our method can be about 20 times faster in encoding and 70-90 times faster in decoding, and our R-D performance is also 2.3% higher. Our method achieves better rate-distortion performance than classical image codecs including H.266/VVC-intra (4:4:4) and some recent learned methods, as measured by both PSNR and MS-SSIM metrics on the Kodak and Tecnick-40 datasets.**

I. Introduction

Recently, deep learning has been successfully applied to the field of image compression, yielding very impressive results. The main components of classical image compression standards, such as JPEG [1], JPEG 2000 [2], BPG (intra-coding of H.265/HEVC) [3], and H.266/VVC [4], include the following key components: linear transform, quantization, and entropy coding. In end-to-end learning-based frameworks, these components have been re-designed carefully.

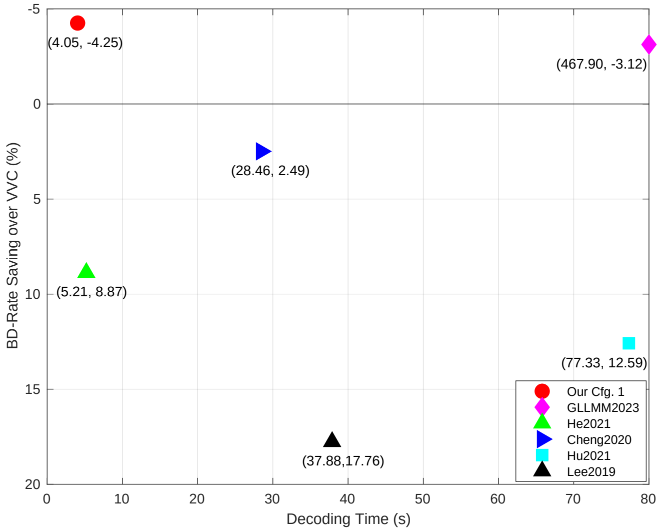


Fig. 1. The decoding time and BD-Rate savings over H.266/VVC for different methods are presented for the Kodak dataset on CPU. A better result is indicated in the upper-left corner. Note that GLLMM [5] has excessive decoding time.

In the transform part, various deep learning-based networks have been developed to extract compact latent representations of the input image, such as residual blocks [6]–[8], attention modules [9], [10], invertible structures [11], or transformer blocks [12], [13]. Although these structures significantly improve the rate-distortion (R-D) performance, their complexity of the networks is usually quite high.

In the quantization part, since learning-based approach requires all components of the codec to be differentiable, but the traditional quantization is not differentiable, different technologies have been proposed to alleviate this problem. For example, in [5], [8], [14]–[16], the quantization is implemented by adding uniform noise to the latent representation during training, and the rounding operation is used during inference.

Haisheng Fu, Feng Liang, Yongqiang Wang and Guohe Zhang are with the School of Microelectronics, Xi'an Jiaotong University, Xi'an, China. (e-mails: fhs4118005070@stu.xjtu.edu.cn; fengliang@xjtu.edu.cn; wangyq0901@stu.xjtu.edu.cn; zhangguohe@xjtu.edu.cn) (Corresponding authors: Feng Liang).

Haisheng Fu, Jie Liang and zhenman Fang are with the School of Engineering Science, Simon Fraser University, Canada (e-mails: hfa23@sfu.ca, jiel@sfu.ca; zhenman@sfu.ca).

Jingning Han is with the Google Inc. (e-mail: jingning@google.com).

This work was supported by the National Natural Science Foundation of China (No. 61474093), Industrial Field Project - Key Industrial Innovation Chain (Group) of Shaanxi Province (2022ZDLGY06-02), the Natural Sciences and Engineering Research Council of Canada (RGPIN-2020-04525), Google Chrome University Research Program, NSERC Discovery Grant RGPIN-2019-04613, DGECR-2019-00120, Alliance Grant ALLRP-552042-2020; CFI John R. Evans Leaders Fund.

For the entropy coding component, the use of the serial context-adaptive entropy model substantially enhances the R-D performance. This is achieved by jointly utilizing hyperpriors and autoregressive models to reduce the spatial redundancy of latent representations. In [17], the transformer-based context model named Contextformer is proposed to further improve the R-D performance. However, these methods cannot be accelerated during the decoding process by parallel computing devices like GPU, making them less suitable for practical applications.

Some recent works using serial context-adaptive entropy model can even outperform the best traditional image standards (i.e. VVC intra coding) in terms of PSNR [5], [11], [18]. In particular, the scheme in [5] represents the current state of the art, where the latent representations are assumed to follow the Gaussian-Laplacian-Logistic mixture model (GLLMM). However, its complexity is quite high.

In this paper, we first propose the deformable residual module to improve the image compression performance. Next, we propose three techniques to reduce the model size and decoding complexity of learned image compression methods while maintaining competitive R-D performance. The main contributions of this paper are summarized as follows:

- We are the first to apply the deformable residual module (DRM) to image compression, which combines the deformable convolution [19] and residual block [20]. The DRM expands the receptive field, making it easier to capture global information. Furthermore, the DRM can reduce the spatial correlation of latent representations, thereby enhancing compression performance.
- Second, we propose an improved checkerboard context model, which divides the latents into two subsets via a checkerboard pattern, and each of them can be processed in parallel, thereby significantly speeding up the decoding. It uses two different networks to estimate the distribution parameters of the two subsets. It also only employs the more powerful GLLMM model in the first subset, because it does not use context model. The second subset only uses the simpler Gaussian mixture model (GMM), without affecting its R-D performance.
- Third, we develop a three-pass knowledge distillation scheme to retrain the encoder, decoder and entropy coding, and reduce the complexity of the core decoder network without sacrificing too much performance. The first pass is to train the teacher network. In the second pass, the student decoder and entropy coding have the same architecture as the teacher network. We jointly train the teacher and student networks to transfer prior information from the teacher to the student, in both the final output and intermediate outputs. In the third pass, we reduce the complexity of the student core decoder network by removing some modules and reducing the number of filters, and retrain the teacher and student networks again.
- Fourth, we introduce L_1 regularization to make the numerical values of the latent representation sparser, and increase the number of zero elements in the latent representation. Then, in the encoding and decoding process, we only encode non-zero channels to save the bit rate, which also reduces the encoding and decoding time.

Experimental results demonstrate that compared to the state-of-the-art (SOTA) learned image coding scheme in [5], our method is approximately 20 times faster in encoding and 70-90 times faster in decoding, and still achieves 2.3% BD-Rate saving. It also offers an attractive trade-off between two other SOTA methods in Wang2023 [21] and Liu2023 [22]. Our method also outperforms the latest classical image codec in H.266/VVC-Intra (4:4:4) and other leading learned schemes such as [16] in both PSNR and MS-SSIM metrics, as shown in Fig. 1.

II. Related Work

Context Models. Most learning-based image compression methods are based on the autoencoder architecture to extract the compact and efficient latent representation of the input image [23]. An autoregressive model is usually used to predict latents from their causal context. In [14], [15], a hyperprior network is introduced to learn some side information to correct the context-based predictions. The data from the context model and the hyper network are then combined to learn the probability distributions of the quantized latents, and guide the entropy coding. In [14], [15], simple Gaussian models are used. In [5], [16], Gaussian Mixture Model (GMM) and Gaussian-Laplacian-Logistic Mixture Models (GLLMM) are proposed to achieve the SOTA performance.

However, serial context models are not friendly to parallel processing during decoding. To address this issue, in [24], a channel-wise autoregressive entropy model is proposed to minimize the element-level serial processing in context model. In [25], a spatial-channel contextual adaptive model is proposed to boost the R-D performance without sacrificing running speed. In [26], a checkerboard context model (CCM) is proposed, which divides all data into two groups in a checkerboard pattern to facilitate parallel processing. However, the R-D performance is dropped by 0.2-0.3 dB on the Kodak dataset.

Deformable Convolution. Dai et al. [19] were the first to utilize deformable convolution in conjunction with learned offset maps to enhance the modeling capability of neural networks. Their method outperformed traditional convolution networks in some challenging vision tasks such as object detection and semantic segmentation. Subsequently, deformable convolution was applied to other computer vision tasks, including action recognition [27] and video super-resolution [28], [29]. In [30], the deformable convolution with dynamic kernels was utilized in video compression to more effectively capture complex non-rigid motion patterns between consecutive frames. This not only boosts motion compensation performance but also reduces the workload for the subsequent residual compression module.

Knowledge Distillation. Knowledge distillation was first proposed in [31], where a lightweight student network is trained to learn the softmax outputs of a trained and complex teacher model. It has been applied in various fields [32]–[36]. The student model distills knowledge by utilizing gradient

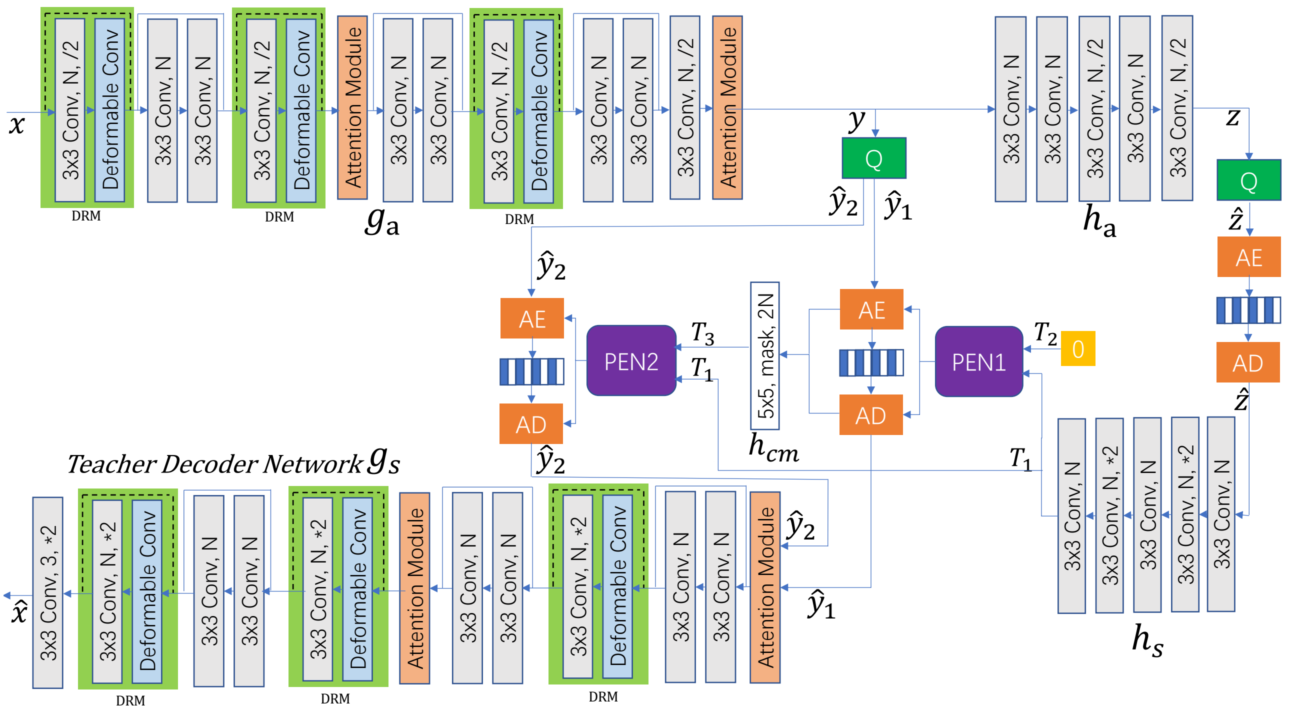


Fig. 2. The architecture of the proposed learned image compression scheme, which includes the core encoder/decoder networks g_a and g_s, and hyperprior networks h_a and h_s. The entire network is used as the teacher network in Fig. 7 to train lower-complexity student network. Deformable Residual Modules (DRM) are used in the core networks. 3×3 indicates the size of the convolution. N is the number of filters. $/2$ and $*2$ represent down/up-sampling operators. Dotted-line shortcut connections indicate changed tensor sizes. AE and AD stand for arithmetic encoder and decoder. The entropy coding is divided into two checkerboard groups. PEN1 and PEN2 are Parameter Estimation Networks for the two groups. h_{cm} is the checkerboard context model network for the second group. More details are in Fig. 3 and Fig. 6.

descent backpropagation of the distillation loss. In [37], the Focal and Global Distillation (FGD) method is proposed to guide the student detector and improves the performance of object detection. In [38], [39], different knowledge distillation methods are designed for image classification and achieve good performance. In [40], the knowledge distillation is first introduced to learned image compression. However, it only focuses on visual performance at low bit rates using the Generative Adversarial Network (GAN). Its network architecture does not include the hyper network, and the performance is thus not very good. Moreover, the distillation process only takes into account the prior knowledge associated with the final output of the teacher network, while the intermediate results of the teacher network are not utilized in distillation.

III. The Proposed Image Compression Framework

In this section, we present the overall architecture of the proposed method. We then introduce the details of the proposed components, which include the improved checkerboard context model, the three-pass knowledge distillation of the decoder and entropy coding, and the corresponding training procedure.

A. The Overall Architecture of the System

The proposed framework is illustrated in Fig. 2. The origin image x has dimensions $W \times H \times 3$, with W and H representing its width and height, respectively. The codec mainly

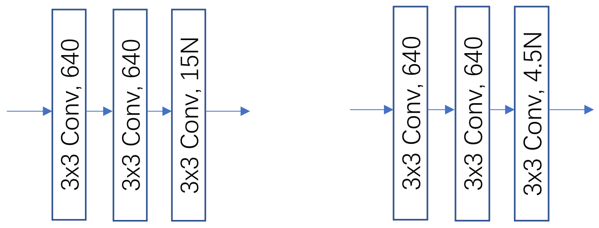


Fig. 3. (a) The architecture of the PEN1 network in Fig. 2. (b) The architecture of the PEN2 network in Fig. 2.

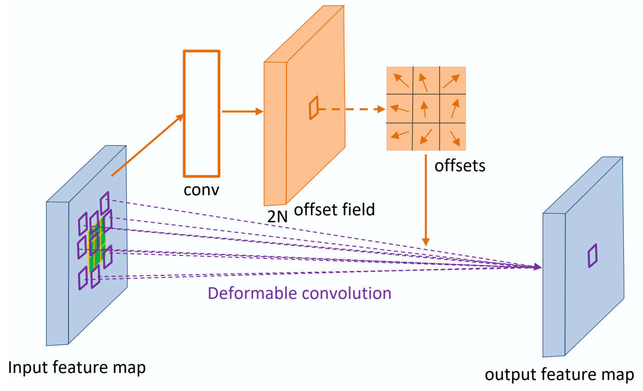


Fig. 4. Illustration of 3×3 deformable convolution.

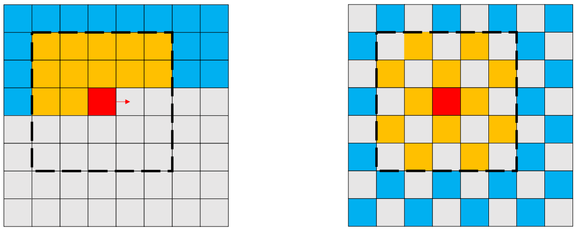


Fig. 5. (a) The serial autoregressive context model. Red cell: the symbol to encode/decode. Orange and blue cells: causal neighbors. Orange cells are examples with a 5×5 convolution window. (b) The checkerboard context model with a 5×5 window. The first pass decodes all blue and orange anchor cells. The second pass decodes all non-anchor cells.

includes the core encoder/decoder networks (g_a and g_s), the hyper networks (h_a and h_s), and the two-step checkerboard-based entropy coding.

The core encoder network g_a learns a compact latent representations y of the input image. g_a is based on that in [16], which includes two simplified attention modules, three residual blocks (shown in gray in Fig. 2), and four stages of pooling operators. We also introduce the deformable residual module (DRM) to the core networks.

To enable parallel entropy decoding of the quantized latents $\hat{y}$, it is divided into two checkerboard subsets $\hat{y}_1$ and $\hat{y}_2$. The probability distribution parameters for the two subsets are estimated by two parameter estimation networks (PENs) via a two-step approach. The details are described in Fig. 3 and Sec. III-C.

Next, arithmetic coding compresses $\hat{y}$ into the output bitstream. The decoded $\hat{y}$ is then fed into the main decoder g_s, which is symmetric to the core encoder network g_a, with convolutions replaced by deconvolutions. Most convolution layers employ the leaky ReLU, with the exception of the final layer in the hyperprior encoder and decoder, which does not use any activation function.

It has been shown in [8] that the complexity of the encoder and decoder networks affects the performance differently. The performance is less sensitive to reduction in the decoder complexity. Similarly, in [41], the authors propose a learning-based method to reduce decoding complexity in neural image compression. It utilizes shallow or linear decoding transforms, complemented by more powerful encoding techniques, aiming to maintain comparable R-D performance while reducing the complexity of the decoding process.

Motivated by the results in [8], [41], in this paper, we develop a three-pass knowledge distillation scheme, which allows us to train a student decoder and entropy coding with lower complexity without sacrificing too much performance. The details are described in Sec. III-D.

B. Deformable Residual Module (DRM)

The deformable convolution was first proposed in [19] and has since been extensively utilized in various domains, including learning-based video compression. The detailed architecture of the deformable convolution is depicted in Fig. 4. Deformable convolution offers benefits by allowing flexible modeling of receptive fields. This helps in extracting better features and representing objects effectively in convolutional neural networks. Consequently, it improves performance in tasks that require precise spatial understanding and object detection. This innovation has the potential to enhance convolutional architectures in capturing complex spatial relationships, making it a promising approach for different computer vision applications.

As depicted in Fig. 4, the dimensions of the offset field align with those of the input feature map, where 2N corresponds to the channel numbers.

In this paper, we propose a deformable residual module (DRM) and apply it to image compression, as depicted in Fig. 2. In our DRM, we combine the deformable module with the classical convolution, and add a shortcut connection. The DRM is used when there are upsampling or downsampling, denoted by dotted-line shortcut in Fig. 2. The DRM can be utilized to reduce spatial redundancy in input image, thereby enhancing image compression performance. In the ablation experiment section, we will demonstrate the effectiveness of this module. As in [19], the deformable module hardly increases the model complexity compared to the classical convolutions.

C. Improved Checkerboard Context Model and Coding

Previous learned image compression methods use serial context-adaptive entropy model. Its decoding cannot be parallelized, as illustrated in Fig. 5(a). To solve this issue, a checkerboard context model is proposed in [26], where the latent representation y is divided into two subsets after quantization, denoted as anchors $\hat{y}_1$ and non-anchors $\hat{y}_2$, as shown in Fig. 5(b). The first pass is to encode and decode the anchors. The second pass is to encode and decode the non-anchors based on anchors. Compared to serial context model used in [16], the decoding of [26] is about $2.5 - 2.7$ times faster.

However, the R-D performance of [26] is dropped by about 0.2-0.3 dB on the Kodak dataset compared to the serial context model used in [16]. There are two reasons for the drop. First, the anchor part is coded using only hyperprior, but without using any context model. Second, a single network is used to estimate the probability distribution parameters of the two subsets.

In this paper, we propose two techniques to improve the R-D performance of the checkerboard-based approach. First, we use two different networks to estimate the probability distribution parameters of the two subsets separately. Next, since the anchor is coded without context model, it should use more powerful probability distribution model to improve the performance. In this paper, we use the more advanced GLLMM model in [5] for the anchor part. The non-anchor part still uses the GMM model, as in [26].

The improved checkerboard context model and decoding are shown in Fig. 2 and Fig. 6. During encoding and training, we first obtain the anchors $\hat{y}_1$ and non-anchors $\hat{y}_2$. In the first pass, we only encode and train the anchors (blue cells in Fig.5(b) and Fig. 6), which only depend on hyperprior and do not adopt

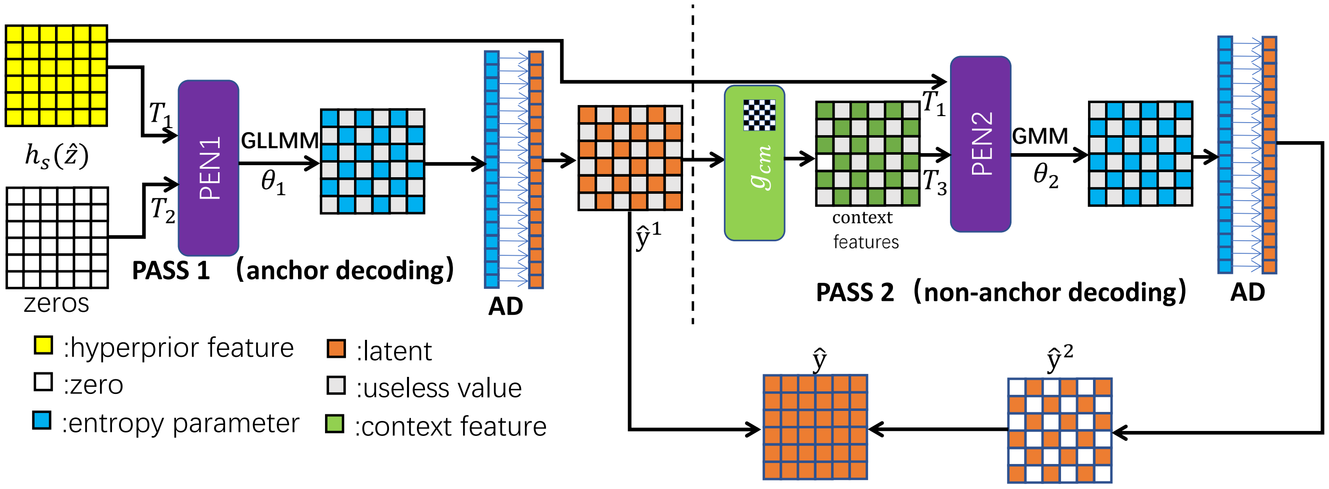


Fig. 6. The details of the proposed two-step checkerboard-based entropy coding scheme.

any context model. The non-anchors $\hat{y}_2$ (grey cells in Fig.5(b) and Fig. 6) are coded using both checkerboard context and the hyperprior.

During decoding, we decode the anchors $\hat{y}_1$ and non-anchors $\hat{y}_2$ in turn, as shown in Fig. 6. $\hat{y}_1$ and $\hat{y}_2$ are initialized to zero tensors, which have the same size as $\hat{y}$. We first utilize the hyper decoder h_s to obtain the output T_1. T_1 and a zero tensor T_2 are first combined and sent to network PEN1 to estimate the probability distribution parameters of the anchors, denoted as θ_1. Different from [26], we use the more powerful GLLMM model in [5] to estimate the parameters of the anchors, to improve the performance even when context model is not used. However, the absence of context model enables us to decode all anchors in parallel.

The decoded anchors are then used to update $\hat{y}_1$, which will pass though a single convolution layer with checkerboard mask (as shown in Fig. 5(b)) to obtain context feature T_3. T_3 is then combined with T_1 and sent to another network PEN2 to estimate the probability distribution parameters of the non-anchors, denoted as θ_2. The non-anchors can also be obtained in parallel. Since PEN1 and PEN2 are trained separately, they can achieve better performance than [26], which only uses one network for both anchors and non-anchors. Since context model is already used for non-anchors, the probability model can be simpler. Therefore only GMM model is used for the non-anchors, as in [26].

The details of the two parameter estimation networks PEN1 and PEN2 are shown in Fig. 3, where as in [5], [16], $15N$ and $4.5N$ are the number of parameters of GLLMM and GMM models respectively.

Finally, we can combine $\hat{y}_1$ and $\hat{y}_2$ to obtain the decoded $\hat{y}$.

D. Three-Pass Knowledge Distillation

In this section, we propose a three-pass knowledge distillation scheme, which allows us to retrain all components in the encoder, decoder and the entropy coding, and then simplify the complexity of the core decoder network g_s in Fig. 2.

In the first pass, we train the entire networks in Fig. 2 using the following traditional loss function:

$$\begin{aligned} L_T &= \lambda_1 D(x,\hat{x}) + H(\hat{y}) + H(\hat{z}) + \lambda_2 L_1(\hat{y}), \\ H(\hat{y}) &= E[-\log_2(P_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}))], \\ H(\hat{z}) &= E[-\log_2(P_{\hat{z}}(\hat{z}))], \end{aligned} \tag{1}$$

where $D(x,\hat{x})$ represents the reconstruction error between the original image x and the reconstructed image $\hat{x}$. We employ the Mean Squared Error (MSE) and MS-SSIM metrics for evaluation in this paper. Additionally, $H(\hat{y})$ and $H(\hat{z})$ denote the entropies associated with the core latent representation and the hyper representation, respectively. L_1 is L_1 norm regularization.

Our goal is to reduce the complexity of the core decoder network g_s, motivated by the results in [8], [41]. However, our experimental results showed that it is not easy to get good performance by reducing the complexity of g_s directly. Therefore in the second pass of our approach, we define a student network, which includes everything in Fig. 2 except for the encoder networks g_a and h_a, as shown in Fig. 7, where the superscripts T and S denote teacher and student, respectively. The initial network architectures of the student network are identical to Fig. 2.

As in other knowledge distillation approaches, we initialize the student network randomly. To pass the knowledge from the teacher network to the student network, in [40], only the prior knowledge related to the final reconstruction image is transferred to the student network. In this paper, we go further by also transferring the prior knowledge in the probability distribution parameters θ_1 and θ_2 of the entropy coding networks to the student network. This helps to train all the decoder and entropy coding components in the student network, including h_s^S, PEN1^S, PEN2^S, h_{cm}^S, and g_s^S.

The teacher and student networks are then jointly retrained using the following loss function, which includes a new knowledge distillation cost.

$$\begin{aligned} L_S &= L_T + \lambda_3 L_{KD}, \\ L_{KD} &= d(\hat{x}^T,\hat{x}^S) + d(\theta_1^T,\theta_1^S)) + d(\theta_2^T,\theta_2^S), \end{aligned} \tag{2}$$

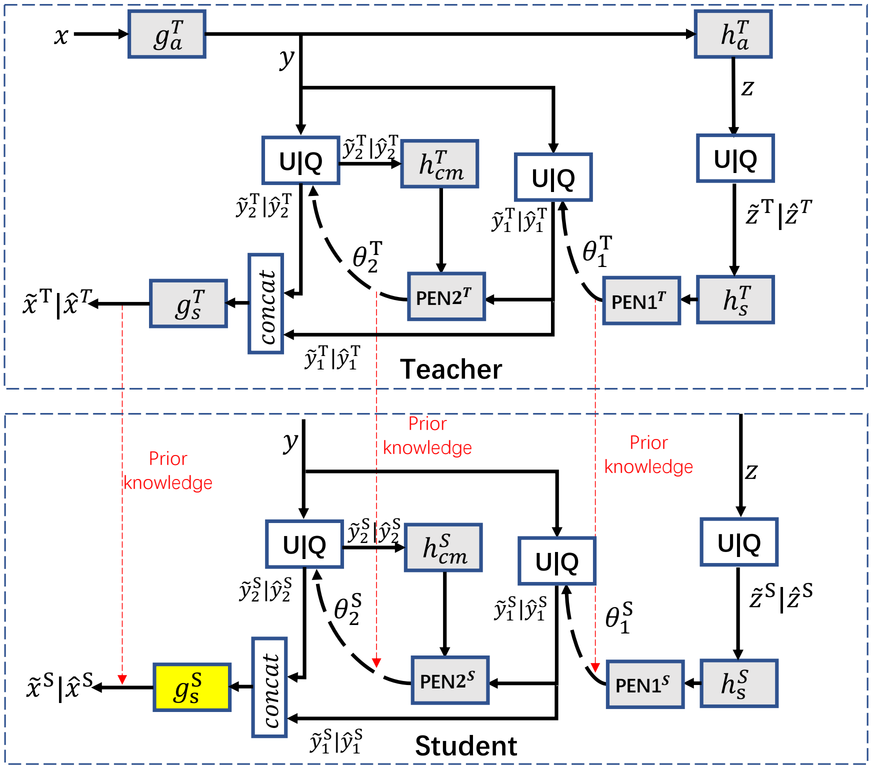


Fig. 7. The knowledge distillation framework between the teacher and student networks. The superscripts S and T stand for Student and Teacher, respectively. All components in the student networks are retrained. The architecture of the core decoder network g_s^S also has lower complexity than g_s^T in the teacher network.

where L_T is the loss function in Eq. 1. L_{KD} is the knowledge distillation cost, which includes the distortions between the teacher and student networks in terms of the reconstructed image, probability distribution parameters θ_1 and θ_2. This ensures the prior knowledge is transferred from the teacher network to the student network.

In [31], [40], softmax is used. In this paper, we find that MSE gives better results, as will be shown in the ablation experiments in Sec. IV.

The results of the second pass allow us to reduce the complexity of the core decoder network g_s^S in the third pass. Starting from the core decoder network in Fig. 2, we reduce its complexity by different approaches, for example, reducing the number of filters N, or removing some modules that have higher complexity but do not contribute too much to the performance, such as the attention modules and residual modules.

To optimize the low-complexity student core decoder network g_s^S, we jointly train the encoder, the simplified student decoder network, and entropy coding again using the joint loss function in Eq. 2.

Note that in both the second and third passes, the encoder is also jointly retrained to ensure the best match with the corresponding decoder.

Experimental results will be presented in Sec. IV to show the performances of various low-complexity core decoder networks.

E. Training Method

Training images are sourced from both the CLIC dataset [44] and the LIU4K dataset [45]. These images are uniformly rescaled to a resolution of 2000×2000. Through the application of data augmentation techniques, including rotation and scaling, we obtain a set of 81,650 training images at a resolution of 384×384.

We optimize our proposed models using two distortion measures: the mean squared error (MSE) and the multi-scale structural similarity (MS-SSIM). For MSE-based optimization, we select λ_1 values from the set $\{0.0016, 0.0032, 0.0075, 0.015, 0.03, 0.045, 0.06\}$. Each selected λ_1 leads to the training of an independent model optimized for a specific bit rate. The number of filters N for the latent representation is configured as 128 for the first three λ_1 values and 256 for the remaining four. For MS-SSIM, λ is assigned the values 12, 40, 80, and 120 sequentially. For λ values of 40 and 80, N is 128, and it increases to 256 for λ values of 80 and 120. Each model undergoes 1.5×10^6 training iterations using the Adam optimization algorithm and a batch size of 8. The initial learning rate is set to 1×10^{-4} for the first 750,000 iterations and subsequently halved after every 100,000 iterations.

The hyperparameters λ_2 and λ_3 are initially set to 0.0001 and 1, respectively. The λ_2 is nullified after the initial 10,000 iterations, and λ_3 is nullified after the first 20,000 iterations. That is, the knowledge distillation is used at the beginning to pass the prior knowledge to the student network. After that,

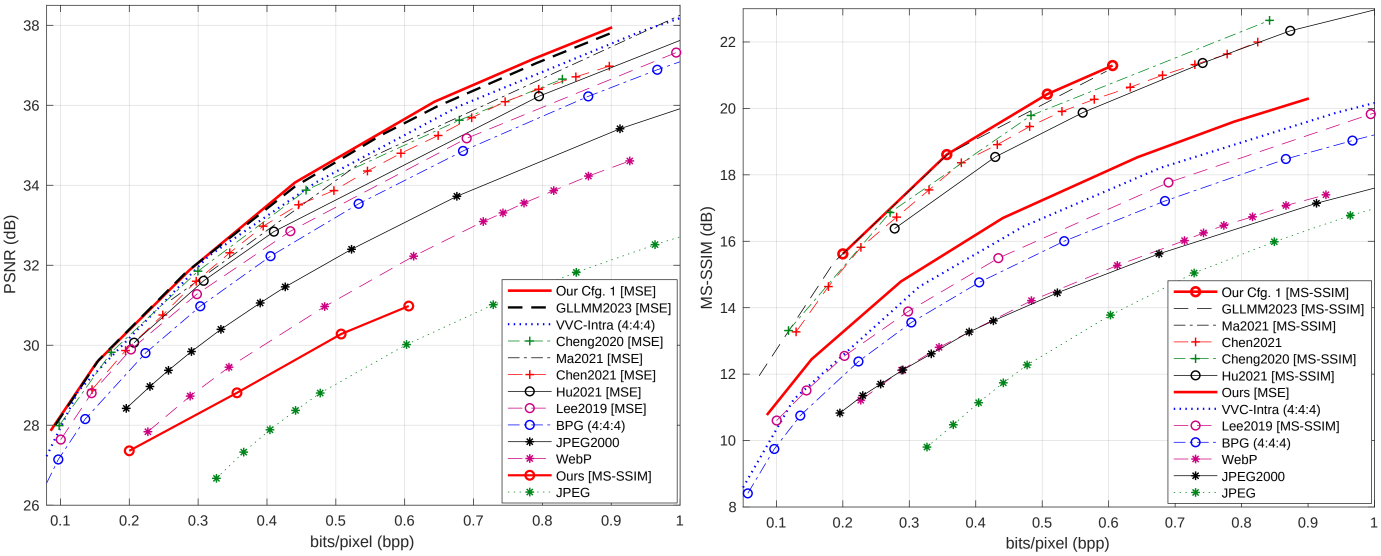


Fig. 8. The R-D curves of different methods in both PSNR and MS-SSIM on the Kodak dataset [42].

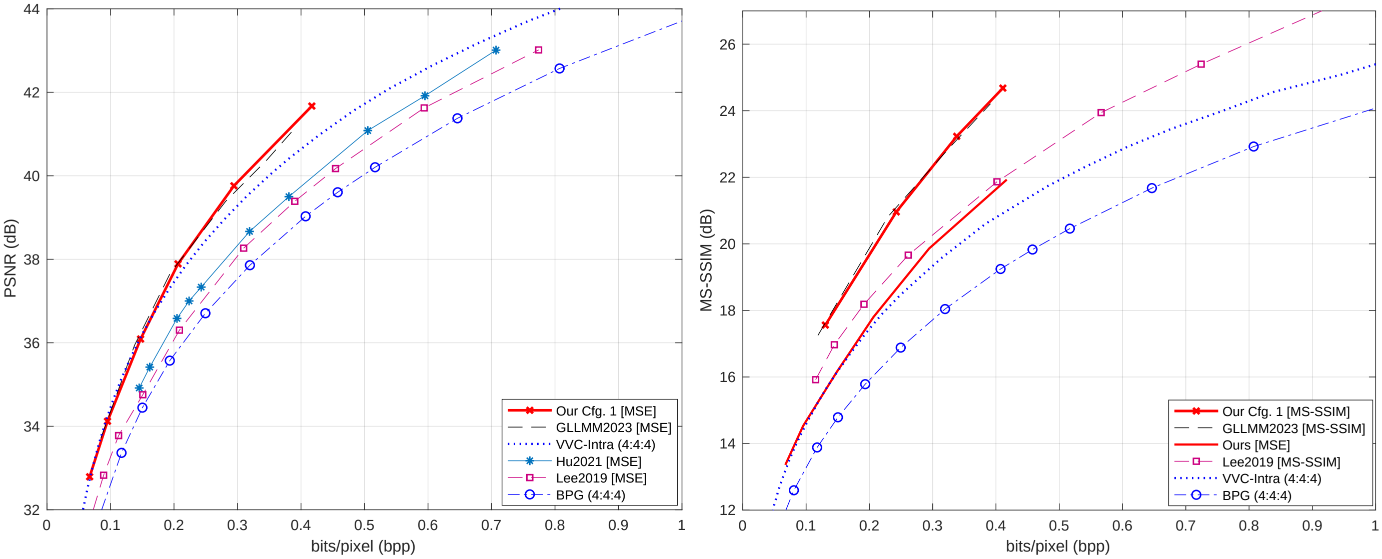


Fig. 9. The R-D curves of different methods in both PSNR and MS-SSIM on the Tecnick-40 dataset [43].

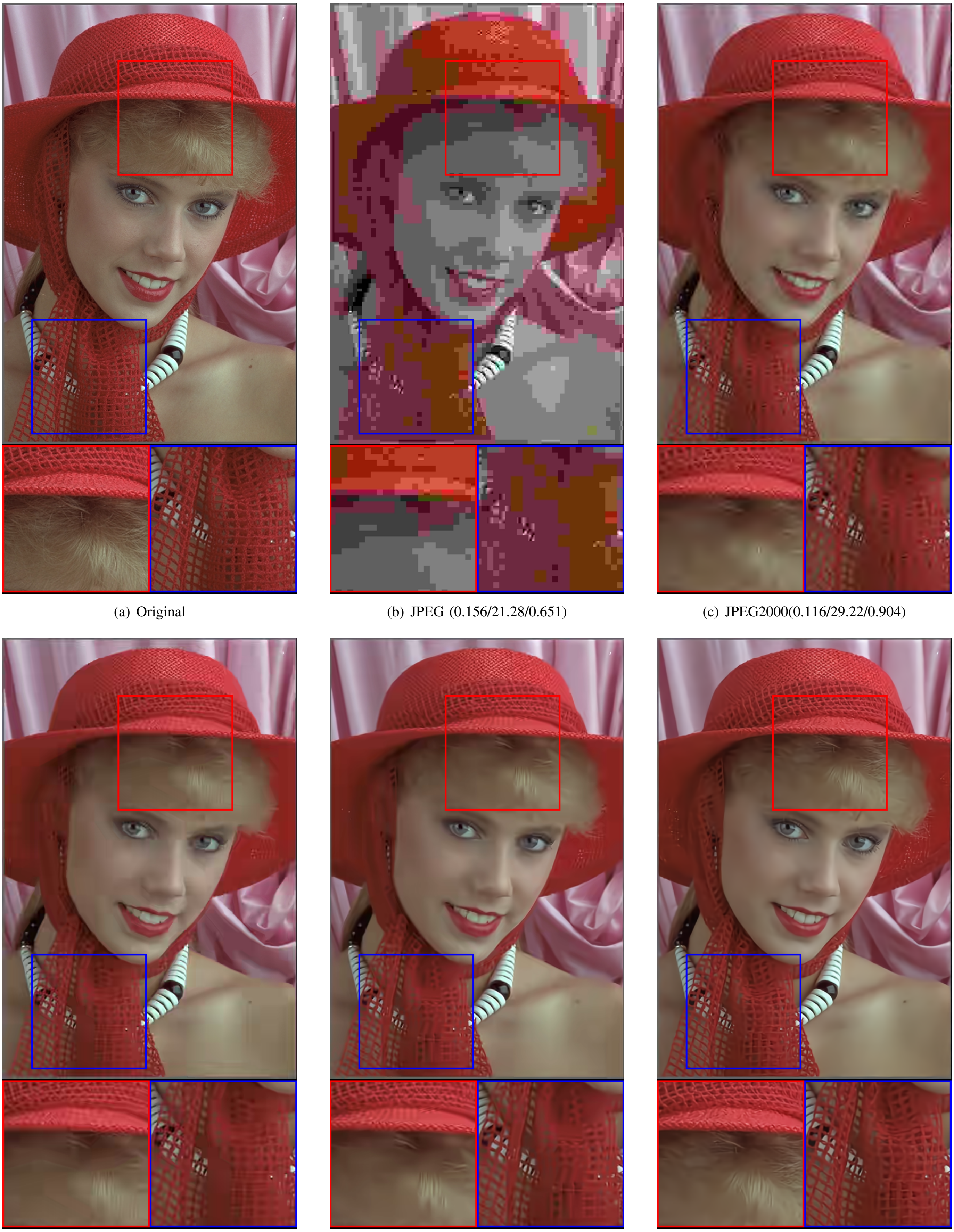
there is no need to have the L_{KD} term in Eq. 2 to reduce the training complexity.

IV. Experimental Results

In this section, we evaluate our proposed method with SOTA learning-based image compression methods as well as traditional methods, using Peak Signal-to-Noise Ratio (PSNR) and MS-SSIM as performance metrics. The evaluation is mainly conducted on two datasets: the Kodak PhotoCD dataset [42], which contains 24 test images with a resolution of 768×512, and the Tecnick-40 dataset [43], comprising 40 test images with a 1200×1200 resolution. We compare our method with learning-based methods such as GLLMM [5], He2021 [26], Hu2020 [46], Cheng2020 [47], Lee2019 [48], Qian2022 [49], Zhu2022 [12], Zou2022 [50], Wang2023 [21], and Liu2023 [22]. We also compare traditional methods including the latest VVC-Intra (4:4:4) [4], BPG-Intra (4:4:4), JPEG2000, and JPEG.

We present our results with four optimized decoder configurations. Cfg. 1 has the teacher decoder architecture as in Fig. 2. Based on Cfg. 1, three low-complexity student networks are trained: Cfg. 2 only removes the attention and residual modules, Cfg. 3 only reduces all N by 25%, and Cfg. 4 only reduces all N by 50%.

Note that for fair comparison, we implement the method in Cheng2020 [47] and increase its number of filters N from 192 to 256 at high rates, which leads to better performance than the original results in [47]. The results of He2021 [26]



(a) Original

(b) JPEG (0.156/21.28/0.651)

(c) JPEG2000(0.116/29.22/0.904)

(d) BPG(0.106/30.02/0.916

(e) VVC(0.103/30.90/0.929)

(f) Ours(0.101/31.05/0.932)

Fig. 10. Visual examples of different image compression methods. Our method is optimized for PSNR. The numbers reported are bit rate (BPP), PSNR (dB), and MS-SSIM.

are based on the source code at [51].

A. R-D Performances

Fig. 8 illustrates the average R-D curves for different methods evaluated on the Kodak dataset. For PSNR-optimized methods, GLLMM (MSE) [5] obtains the best performance among the competing methods, which also outperforms VVC (4:4:4). Our Cfg. 1 obtains almost the same coding performance as GLLMM at low bit rates and has better performance at high bit rates. Cfg. 1 achieves the same performance with VVC (4:4:4) at low bit rates. When the bit rate is higher than 0.4 bpp, Cfg. 1 has a gain of 0.25-0.3 dB over VVC (4:4:4). For MS-SSIM, our method is also slightly higher than GLLMM. A visual example is given in Fig. 10.

Fig. 9 presents the results on the Tecnick-40 dataset. Our Cfg. 1 achieves the same performance as GLLMM at bit rates below 0.2 bpp and is sightly better than GLLMM [5] at rates higher than 0.2 bpp. Additionally, Cfg. 1 outperforms other learned image compression methods and all traditional image codecs.

B. Complexity and Performance Trade-offs

Table I presents a comparative analysis of average encoding/decoding times, BD-Rate savings over VVC [53], and model sizes at low and high bit rates across different methods. Since GLLMM [5], VVC, Hu2020 [46], and Cheng2020 [16] suffer from a non-determinism issue on GPU and only run on CPU [54], we evaluate different methods on the same CPU (2.9GHz Intel Xeon Gold 6226R CPU). We use Python's Time library functions to measure encoding and decoding times.

Compared to the state-of-the-art GLLMM [5], our Cfg. 1 has improved encoding speed by approximately 20 times and decoding speed by 70-90 times. Our model even achieves better R-D performance and has a smaller size than GLLMM.

We can conclude from Fig. 8, Fig. 9, and Table I that the proposed scheme outperforms GLLMM in both R-D performance and complexity.

Compared to Cheng2020 [16], our Cfg. 1 encoding time is similar, but decoder is about 4-5 times faster. Our R-D performance is 6.85% and 11.20% better. Our speed is similar to [26], but our R-D performance is about 15% better.

Our Cfg. 2 and Cfg. 3 can further reduce the decoder complexity by $20-30\%$, with $2.6-4.0\%$ loss in R-D performance compared to Cfg. 1, but still have better performance than other learning-based methods and VVC (4:4:4). Cfg. 4 is faster but has 18.3% drop in R-D performance. Therefore our method can offer various trade-offs between complexity and R-D performance.

To further evaluate the performance of our method, we compare our Cfg. 1 with some recent methods on the Kodak dataset and the same GPU (NVIDIA Tesla V100 with 16 GB memory). These methods include Zhu2022 [12], Qian2022 [49], Zou2022 [50], Wang2023 [21], and Liu2023 [22]. These methods do not use the causal entropy model, thus allowing for parallel acceleration on the GPU. We use Python's Time library functions to measure encoding and decoding times. We also evaluate the model parameters and computational complexity using the PyTorch Flops Profiler tool [1].

As shown in Table II, our method is better than Zhu2022, Qian2022, and Zou2022 in both complexity and R-D performance. Wang2023 has the lowest complexity, but BD-rate saving is 3.3% worse than ours. Liu2023 has 2.24% more BD-rate saving, but model size is 3.2 times of ours. Our method achieves a trade-off between Wang2023 and Liu2023 in complexity and performance.

C. Ablation Experiments

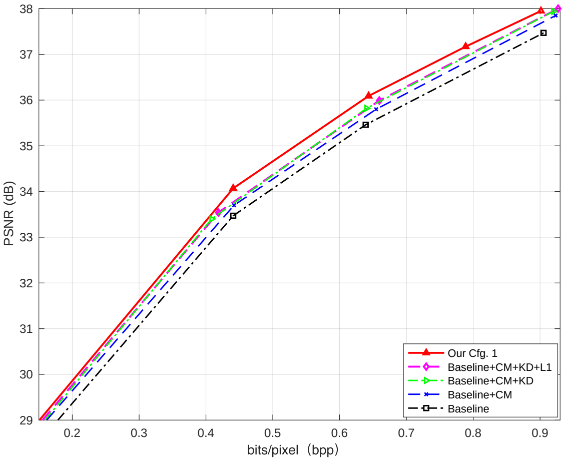


Fig. 11. The contributions of the improved checkerboard context model and the knowledge distillation for Kodak dataset.

In this part, we present various ablation experiments.

1) Contributions of the Proposed Modules: We first show the contributions of the improved checkerboard context model, the knowledge distillation, L_1 regularization, and DRM in Fig. 11 for the Kodak dataset. We replace the GMM model in [16] with the checkerboard entropy model [26], and other components remain unaltered. The revised scheme is adopted as the baseline. On top of the baseline, we sequentially add different modules.

We first replace the checkerboard entropy model [26] in the baseline by the proposed checkerboard entropy model [26], denoted as Baseline+CM, which improves the coding performance by about 0.2-0.3 dB at the same bit rate. Next, we add the knowledge distillation to Baseline+CM, denoted as Baseline+CM+KD. Compared to Baseline+CM, Baseline+CM+KD improves the coding performance by about 0.1-0.15 dB at the same bit rate. Then, we add the L_1 regularization to the loss function, denoted as Baseline+CM+KD+L1. It can be observed that introducing L_1 regularization does not reduce the encoding performance. It just makes the latent representations more sparse. Last, we add the DRM to Baseline+CM+KD+L1, which is our proposed method. Compared to Baseline+CM+KD+L1, the proposed full method will improve the performance by another 0.1-0.15 dB.

[1] https://www.deepspeed.ai/tutorials/flops-profiler

TABLE I
COMPARISONS OF ENCODING AND DECODING TIME, BD-RATE SAVING OVER VVC, AND MODEL SIZES ON CPU.

Dataset	Method	Encoding time	Decoding time	BD-Rate	Model size(Low)	Model size(High)
Kodak	VVC	402.27s	0.607s	0.0	7.2 MB	7.2MB
	Lee2019 [48]	10.721s	37.88s	17.0%	123.8 MB	292.6MB
	Hu2021 [52]	35.7187s	77.3326s	11.1 %	84.6 MB	290.9MB
	Cheng2020 [16]	26.37s	28.46s	2.6 %	50.8 MB	175.18MB
	He2021 [26]	24.36s	5.21s	8.9 %	46.6 MB	156.6 MB
	GLLMM [5]	467.90s	467.90s	-3.13%	77.08 MB	241.03MB
	Our Cfg. 1	**25.08 s**	**4.45s**	**-4.25%**	**63.06 MB**	**197.8MB**
	Our Cfg. 2	**24.02 s**	**3.03s**	**-1.89%**	**54.26 MB**	**166.9MB**
	Our Cfg. 3	**22.56 s**	**2.78s**	**-0.19%**	**54.66 MB**	**164.1MB**
	Our Cfg. 4	**18.24 s**	**2.45s**	**14.23%**	**47.6 MB**	**134.1MB**
Tecnick	VVC	700.59s	1.49s	0.0	7.2 MB	7.2MB
	Lee2019 [48]	54.8s	138.81s	31.59 %	123.8 MB	292.6MB
	Hu2021 [52]	84.035s	271.50s	23.06 %	84.6 MB	290.9MB
	Cheng2020 [16]	59.48s	71.71s	5.93 %	50.8MB	175.18MB
	He2021 [26]	56.26s	12.45s	12.21 %	46.6 MB	156.6 MB
	GLLMM [5]	1233.05s	1245.05s	-5.14%	77.08 MB	241.03MB
	Our Cfg. 1	**57.63**s	**11.65s**	**-5.27%**	**63.06 MB**	**197.8 MB**
	Our Cfg. 2	**50.47**s	**7.65s**	**-2.38%**	**54.26 MB**	**166.9 MB**
	Our Cfg. 3	**46.56**s	**5.23s**	**-1.20%**	**54.66 MB**	**164.1 MB**
	Our Cfg. 4	**38.67**s	**4.78 s**	**15.78%**	**47.6 MB**	**134.1 MB**

TABLE II
COMPARISON OF ENCODING AND DECODING TIMES, BD-RATE, AND PARAMETERS FOR KODAK DATASET ON GPU.

Methods	Enc. Time	Dec. Time	BD-rate	# Param.
Zhu2022 [12]	0.269s	0.183s	-3.88%	32.34 MB
Qian2022 [49]	4.78s	4.78s	3.15%	128.86 MB
Zou2022 [50]	0.163s	0.184s	-2.22%	99.86 MB
Wang2023 [21]	0.065s	0.045s	-0.95%	9.86 MB
Liu2023 [22]	0.182s	0.212s	-6.49%	45.18 MB
Ours	0.154s	0.188s	-4.25%	14.60 MB

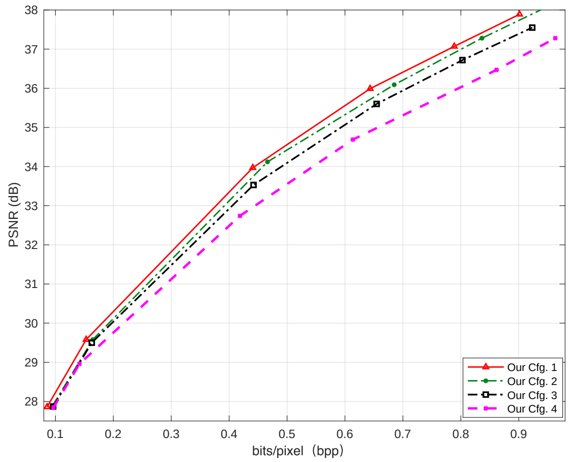


Fig. 12. R-D performances of different configurations of the proposed method for Kodak dataset.

2) Comparison of Four Configurations: Fig. 12 shows the detailed R-D curves of the four configurations of our method for the Kodak dataset. Together with Table I, it can be observed that the R-D performances of Cfg. 2 and Cfg. 3 are only slightly lower than Cfg. 1, and the model size is reduced by about 15%. The PSNR of Cfg. 4 is more than 1 dB lower than Cfg. 1 at high rates, which shows that at high rates, the network needs more filters to ensure good performance. These results suggest that we can combine different knowledge distillation methods. For example, at low bit rates, we can reduce the number filters. At high bit rates, we can remove the attention models and residual blocks.

3) Comparison of Different Loss functions: We introduce the L_1 regularization to make the latent representation more sparse. Our method leads to a higher frequency of zeros and increases the probability of skipping all-zero channels. Table III compares the number of all-zero channels, total channels, decoding times (both with and without skipping all-zero channels), and the decoding time reduction rate achieved by skipping all-zero channels. Notably, our approach results in a 48-59% reduction in decoding time.

An illustrative example from the Kodak dataset is shown in Fig. 13. It can be observed that the introduction of the L1 regularization into the loss function results in a sparser latent representation.

Table IV compares the performance when the Softmax and MSE are used in the knowledge distillation loss function L_{KD}, which shows that MSE has better performance.

4) Comparison of Different Modules: We conducted some experiments to compare our DRM with other techniques, including transformers [55], swin transformers [56], non-local attention modules [16], and the traditional convolution, all at the same bit rate on the Kodak dataset. We ensured that all configuration parameters were identical during training. The results are shown in Fig. 14 and Table V for Kodak dataset.

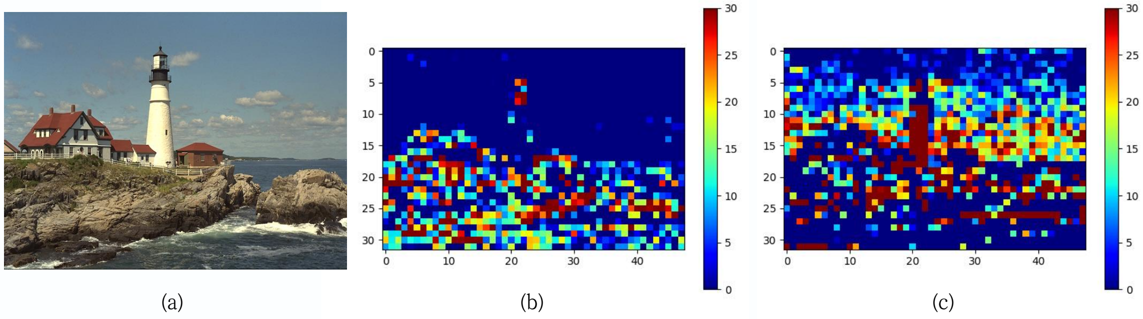
As shown in Fig. 14, our proposed DRM achieves better performance than the other modules. The Swin Transformer [56] and the non-local attention module [16] are about 0.1 dB lower than our method. Transformer is about 0.3 dB lower than ours.

Table V shows that compared to the traditional convolution, the DRM only increases the number of parameters by 1.4%. Our number of parameters is also lower than transformer and attention methods.

5) Further Comparisons on Other Datasets: We also conduct additional experiments to compare our method with the

TABLE III
DECODING TIME OF OUR METHOD WITH AND WITHOUT SKIPPING ALL-ZERO CHANNELS FOR KODAK DATASET.

Name	Bit rates	All-Zero Channels	Total Channels	Dec. Time (Ours)	Dec. Time (Full)	Dec. Reduction
Kodak	Low	76	128	4.35 s	6.45s	48.27%
Kodak	High	124	256	64.43 s	100.22 s	55.54%
Tecnick	Low	78	128	11.65 s	17.44 s	49.37 %
Tecnick	High	123	256	203.49 s	324.58 s	59.50%



(a)

(b)

(c)

Fig. 13. (a) An original image in the Kodak dataset. (b) The average latent representations with L_1 regularization in our method. (c) The average latent representations without L_1 regularization in our method.

TABLE IV
COMPARISON OF USING DIFFERENT KNOWLEDGE DISTILLATION LOSSES IN OUR METHOD FOR KODAK DATASET.

Module	Bit rate	PSNR (dB)	MS-SSIM (dB)
Softmax	0.1643	29.67	12.60
MSE	**0.1628**	**29.76**	**12.62**
Softmax	0.8046	37.05	19.58
MSE	**0.8028**	**37.23**	**19.68**

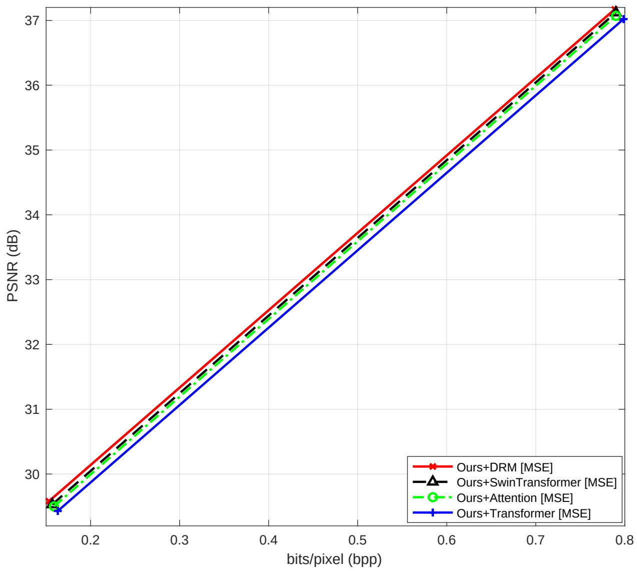


Fig. 14. R-D performances of different advanced modules on Kodak dataset.

TABLE V
COMPARISON OF DIFFERENT MODULES ON ENCODING TIME, DECODING TIME, AND PARAMETERS FOR KODAK DATASET.

Scheme	Enc. Time	Dec. Time	# Paras
Ours+Attention	0.165s	0.145s	16.43 MB
Ours+Transformer	0.163s	0.184s	16.80 MB
Ours+Swin Transformer	0.182s	0.212s	16.18 MB
Ours+Convolution	0.132s	0.169s	14.41 MB
Ours+DRM	0.154s	0.188s	14.60 MB

two recent methods in Wang2023 [21] and Liu2023 [22] using the Kodak, CLIC [57], and COCO [58] datasets. The CLIC 2021 test set [59] includes 60 images with resolutions ranging from 751×500 to 2048×2048. We randomly selected 30 test images from the COCO 2018 test set, with resolutions ranging from 320×240 to 624×640. We consider VVC as our baseline when computing the BD-rate metric. These tests were conducted on a NVIDIA Tesla V100 with 12 GB of memory.

The test result are shown in Table VI. Compared to Wang2023, our method has better R-D performance (up to 21% for COCO), but our model size is increased by about 15%, and the #KMACs is about 5 times (mainly due to the GLLMM module). Compared to Liu2023, our R-D performance is 2% lower, but our model size is only about 1/3, with similar #KMACs. Therefore our method offers a trade-off between these two methods.

V. CONCLUSIONS

In this paper, we propose four techniques to improve the R-D performance of learned image compression and reduces its complexity, based on deformable residual module (DRM), improved checkerboard context model, knowledge distillation, and L_1 regularization respectively. We introduce the DRM to further reduce the spatial redundancy of latent representations and improve the coding performance. In the checkerboard context model, we use a two-step checkerboard entropy coding to estimate the probability distribution parameters of the two subsets. We only employ the GLLMM model for the first subset, which does not use context model. The second subset

TABLE VI
COMPARISONS OF ENCODING AND DECODING TIME, BD-RATE SAVING OVER VVC, AND MODEL SIZES.

Dataset	Method	Encoding time	Decoding time	BD-Rate	#Paras	#KMACs
Kodak	Wang2023 [21]	0.065s	0.045s	-0.95%	12.78 MB	41.67GMACs
	Liu2023 [22]	0.182s	0.212s	-6.49%	45.18 MB	215.32 GMACs
	Ours	0.154s	0.188s	-4.25 %	14.60 MB	210.53GMACs
CLIC2021	Wang2023 [21]	0.089s	0.084s	10.86%	12.78 MB	235.6GMACs
	Liu2023 [22]	0.787s	0.884s	-7.634%	45.18 MB	1.15TMACs
	Ours	0.756s	0.867s	-5.628 %	14.60 MB	1.12TMACs
COCO2018	Wang2023 [21]	0.048s	0.031s	21.61%	12.78 MB	34.73 GMACs
	Liu2023 [22]	0.185s	0.204s	-3.27%	45.18 MB	179.43 GMACs
	Ours	0.135s	0.168s	-1.03 %	14.60 MB	210.53GMACs

only uses the simpler GMM, but uses the first subset for context model. We develop a three-pass knowledge distillation scheme to retrain the encoder, decoder, and entropy coding, and also reduce the complexity of the core decoder network. We introduce L_1 regularization to make the latent representation more sparse, and only encode and decode non-zero channels, which greatly reduces the encoding and decoding time without sacrificing coding performance.

Extensive experimental results demonstrate that our proposed method achieves better R-D performance than a SOTA method in GLLMM [5], and is 70-90 times faster. It also offers an attractive trade-off between two other SOTA methods in Wang2023 [21] and Liu2023 [22]. Our method also has better performance than traditional image codecs including the H.266/VVC in terms of both PSNR and MS-SSIM metrics.

The checkerboard context model and knowledge distillation proposed in this paper can be further optimized in the future.

REFERENCES

[1] G. K. Wallace, "The jpeg still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18–34, 1992.

[2] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The jpeg 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. 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, 2012.

[4] H. Fraunhofer, "Vvc official test model vtm," 2019. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tree/VTM-5.2

[5] H. Fu, F. Liang, J. Lin, B. Li, M. Akbari, J. Liang, G. Zhang, D. Liu, C. Tu, and J. Han, "Learned image compression with gaussian-laplacian-logistic mixture model and concatenated residual modules," *IEEE Transactions on Image Processing*, vol. 32, pp. 2063–2076, 2023.

[6] H. Chen, X. He, H. Yang, L. Qing, and Q. Teng, "A feature-enriched deep convolutional neural network for jpeg image compression artifacts reduction and its applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 1, pp. 430–444, 2022.

[7] H. Fu, F. Liang, B. Lei, N. Bian, Q. Zhang, M. Akbari, J. Liang, and C. Tu, "Improved hybrid layered image compression using deep learning and traditional codecs," *Signal Processing: Image Communication*, vol. 82, p. 115774, 2020.

[8] H. Fu, F. Liang, J. Liang, B. Li, G. Zhang, and J. Han, "Asymmetric learned image compression with multi-scale residual block, importance scaling, and post-quantization filtering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 4309–4321, 2023.

[9] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, "End-to-end learnt image compression via non-local attention optimization and improved context modeling," *IEEE Transactions on Image Processing*, vol. 30, pp. 3179–3191, 2021.

[10] M. Li, K. Zhang, J. Li, W. Zuo, R. Timofte, and D. Zhang, "Learning context-based nonlocal entropy modeling for image compression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 3, pp. 1132–1145, 2023.

[11] Y. Xie, K. L. Cheng, and Q. Chen, "Enhanced invertible encoding for learned image compression," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 162–170.

[12] Y. Zhu, Y. Yang, and T. Cohen, "Transformer-based transform coding," in *International Conference on Learning Representations*, 2022.

[13] R. Zou, C. Song, and Z. Zhang, "The devil is in the details: Window-based attention for image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17 492–17 501.

[14] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018, pp. 1–23.

[15] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 794–10 803.

[16] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7939–7948.

[17] A. B. Koyuncu, H. Gao, A. Boev, G. Gaikov, E. Alshina, and E. Steinbach, "Contextformer: A transformer with spatio-channel attention for context modeling in learned image compression," in *Computer Vision – ECCV 2022*, 2022, pp. 447–463.

[18] J. Lee, S. Cho, and M. Kim, "Joint autoregressive and hierarchical priors for learned image compression," *arXiv:1912.12817*, 2020.

[19] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 764–773.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[21] G.-H. Wang, J. Li, B. Li, and Y. Lu, "Evc: Towards real-time neural image compression with mask decay," 2023.

[22] J. Liu, H. Sun, and J. Katto, "Learned image compression with mixed transformer-cnn architectures," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 14 388–14 397.

[23] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations*, 2017.

[24] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3339–3343.

[25] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 5718–5727.

[26] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 14 771–14 780.

[27] J. P. Klopp, L.-G. Chen, and S.-Y. Chien, "Utilising low complexity cnns to lift non-local redundancies in video coding," *IEEE Transactions on Image Processing*, vol. 29, pp. 6372–6385, 2020.

[28] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "Tdan: Temporally-deformable alignment network for video super-resolution," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3357–3366.

[29] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. C. Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *2019*

IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1954–1963.

[30] Z. Hu, D. Xu, G. Lu, W. Jiang, W. Wang, and S. Liu, "Fvc: An end-to-end framework towards deep video compression in feature space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4569–4585, 2023.

[31] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[32] H. Chen, Y. Wang, H. Shu, C. Wen, C. Xu, B. Shi, C. Xu, and C. Xu, "Distilling portable generative adversarial networks for image translation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3585–3592, Apr. 2020.

[33] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=lL3lnMbR4WU

[35] S. Li, M. Lin, Y. Wang, Y. Wu, Y. Tian, L. Shao, and R. Ji, "Distilling a powerful student model via online knowledge distillation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2022.

[36] Y. Yu, B. Li, Z. Ji, J. Han, and Z. Zhang, "Knowledge distillation classifier generation network for zero-shot learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 6, pp. 3183–3194, 2023.

[37] Z. Yang, Z. Li, X. Jiang, Y. Gong, Z. Yuan, D. Zhao, and C. Yuan, "Focal and global knowledge distillation for detectors," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2022, pp. 4633–4642.

[38] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1365–1374.

[39] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1921–1930.

[40] L. Helminger, R. Azevedo, A. Djelouah, M. Gross, and C. Schroers, "Microdosing: Knowledge distillation for gan based compression," 2022.

[41] Y. Yang and S. Mandt, "Computationally-efficient neural image compression with shallow decoders," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 530–540.

[42] *Kodak PhotoCD dataset, http://r0k.us/graphics/kodak/*. [Online]. Available: http://r0k.us/graphics/kodak/

[43] *Tecnick dataset, https://bellard.org/bpg/*. [Online]. Available: https://bellard.org/bpg/

[44] *CLIC dataset, http://www.compression.cc/*. [Online]. Available: http://www.compression.cc/

[45] J. Liu, D. Liu, W. Yang, S. Xia, X. Zhang, and Y. Dai, "A comprehensive benchmark for single image compression artifact reduction," *IEEE Transactions on Image Processing*, vol. 29, pp. 7845–7860, 2020.

[46] Y. Hu, W. Yang, and J. Liu, "Coarse-to-fine hyper-prior modeling for learned image compression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 013–11 020.

[47] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Energy compaction-based image compression using convolutional autoencoder," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 860–873, 2020.

[48] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," in *International Conference on Learning Representations*, 2019.

[49] Y. Qian, X. Sun, M. Lin, Z. Tan, and R. Jin, "Entroformer: A transformer-based entropy model for learned image compression," in *International Conference on Learning Representations*, 2022.

[50] R. Zou, C. Song, and Z. Zhang, "The devil is in the details: Window-based attention for image compression," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 471–17 480.

[51] M. Lu and Z. Ma, "High-efficiency lossy image coding through adaptive neighborhood information aggregation," *arXiv preprint arXiv:2204.11448*, 2022.

[52] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning end-to-end lossy image compression: A benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[53] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," 2001, VCEG-M33.

[54] H. Sun, L. Yu, and J. Katto, "Learned image compression with fixed-point arithmetic," in *2021 Picture Coding Symposium (PCS)*, 2021, pp. 1–5.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[56] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

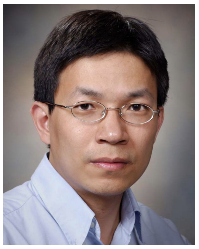
[57] *The 2019 Workshop and Challenge on Learned Image Compression (CLIC),http://www.compression.cc/*, 2018. [Online]. Available: http://www.compression.cc/

[58] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, 2014, pp. 740–755.

[59] G. Toderici, R. Timofte, J. Balle, E. Agustsson, N. Johnston, and F. Mentzer, "2021 workshop and challenge on learned image compression (clic)." [Online]. Available: http://www.compression.cc

Haisheng Fu (Student Member, IEEE) received the Ph.D. degree in electronic science and technology from Xi'an Jiaotong University, Xi'an. During his Ph.D. studies, he spent two years as a joint Ph.D. student at Simon Fraser University, Canada. He is currently a postdoctoral fellow at Simon Fraser University. His research interests include Machine Learning, Image and Video Compression, Deep Learning, and VLSI Design. He is also an active reviewer for several prestigious journals and conferences, including IEEE TPAMI, IEEE TIP, IEEE TCSVT, ICASSP, and ICIP.

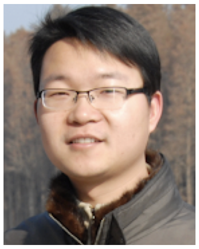
Feng Liang is currently Professor of the Microelectronics School at Xi'an Jiaotong University. He earned his B.E. from Zhengzhou University and his M.E. and Ph.D. from Xi'an Jiaotong University. His current research interests include Signal Processing, Machine Learning, VLSI design, CIM, and computer architecture.



Jie Liang (Senior Member, IEEE) received the B.E. and M.E. degrees from Xi'an Jiaotong University, China, the M.E. degree from National University of Singapore, and the PhD degree from the Johns Hopkins University, USA, in 1992, 1995, 1998, and 2003, respectively. From 2003 to 2004, he worked at the Video Codec Group of Microsoft Digital Media Division. Since May 2004, he has been with the School of Engineering Science, Simon Fraser University, Canada, where he is currently a Professor.

Jie Liang's research interests include Image and Video Processing, Computer Vision, and Deep Learning. He had served as an Associate Editor for several journals, including IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), and IEEE Signal Processing Letters. He has also served on three IEEE Technical Committees. He received the 2014 IEEE TCSVT Best Associate Editor Award, 2014 SFU Dean of Graduate Studies Award for Excellence in Leadership, and 2015 Canada NSERC Discovery Accelerator Supplements (DAS) Award.

YongQiang Wang was born in Gansu, China. He received the B.E. degree in Central South University. He is currently pursuing the Ph.D. degree in electronic science and technology from Xi'an Jiaotong University, Xi'an. His research interests include Deep Learning, Image and Video compression.



Zhenman Fang (Member, IEEE) received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2014. He conducted postdoctoral research at the University of California, Los Angeles (UCLA), Los Angeles, CA, USA, from 2014 to 2017, and worked as a Staff Software Engineer at Xilinx, San Jose, CA, USA, from 2017 to 2019. He is currently an Assistant Professor with the School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada. His recent research focuses on customizable computing with specialized hardware acceleration, including emerging application characterization and acceleration, novel accelerator-rich and near-data computing architecture designs, and corresponding programming, runtime, and tool support.

Guohe Zhang received the B.S. and Ph.D. degrees in electronics science and technology from Xi'an Jiaotong University, Shaanxi, China, in 2003 and 2008, respec- tively. He is currently an Associate Professor with the School of Microelectronics, Xi'an Jiaotong University. In 2009, he joined the School of Elec- tronic and Information Engineering, as a Lecturer. He was promoted to an Associated Professor, in 2013. From 2009 to 2011, he had a three year's Postdoctoral Researcher with the School of Nuclear Science and Technology, Xi'an Jiaotong University. From February to May of 2013, he had a short term visiting to the University of Liverpool, U.K. His research interests fall in the area of semiconductor device physics and modeling, VLSI design and testing.

Jingning Han (Senior Member, IEEE) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2008 and 2012, respectively. He joined the WebM Codec Team, Google, Mountain View, CA, USA, in 2012, where he is the Main Architect of the VP9 and AV1 codecs, and leads the Software Video Codec Team. He has published more than 60 research articles. He holds more than 50 U.S. patents in the field of video coding. His research interests include video coding and computer science architecture. Dr. Han received the Dissertation Fellowship from the Department of Elec- trical and Engineering, University of California at Santa Barbara, in 2012. He was a recipient of the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo, in 2012. He also received the IEEE Signal Processing Society Best Young Author Paper Award, in 2015.