Learned image compression with transformers

Tianma Shen, Ying Liu

Tianma Shen, Ying Liu, "Learned image compression with transformers," Proc. SPIE 12522, Big Data V: Learning, Analytics, and Applications , 1252207 (13 June 2023); doi: 10.1117/12.2656516

Event: SPIE Defense + Commercial Sensing, 2023, Orlando, Florida, United States

Learned Image Compression with Transformers

Tianma Shen and Ying Liu

Department of Computer Science and Engineering, Santa Clara University
Santa Clara, CA 95053, United States

ABSTRACT

Recent years have witnessed great advances in deep learning-based image compression, also known as learned image compression. An accurate entropy model is essential in learned image compression, since it can compress high-quality images with a lower bit rate. Current learned image compression schemes developed entropy models using context models and hyperpriors. Context models utilize local correlations within latent representations for better probability distribution approximation, while hyperpriors provide side information to estimate distribution parameters. Most recently, several transformer-based learned image compression algorithms have emerged and achieved state-of-the-art rate distortion performances, surpassing existing convolutional neural network (CNN)-based learned image compression and traditional image compression. Transformers are better at modeling long-distance dependencies and extracting global features than CNNs. However, the research of transformer-based image compression is still in its early stage. In this work, we propose a novel transformer-based learned image compression model. It adopts transformer structures in the main image encoder and decoder and in the context model. In particular, we propose a transformer-based spatial-channel auto-regressive context model. Encoded latent-space features are split into spatial-channel chunks, which are entropy encoded sequentially in a channel-first order, followed by a 2D zigzag spatial order, conditioned on previously decoded feature chunks. To reduce the computational complexity, we also adopt a sliding window to restrict the number of chunks participating in the entropy model. Experimental studies on public image compression datasets demonstrate that our proposed transformer-based learned image codec outperforms traditional image compression and existing learned image compression models visually and quantitatively.

Keywords: Context model, deep learning, entropy model, hyperprior, image compression, learned image compression, transformer

1. INTRODUCTION

Image compression plays an important role in media storage and transmission, in which the main goal is to obtain high-quality images at a given compression rate or bit rate. Analysis transform, synthesis transform, quantization, and entropy coding are the most crucial parts of image compression. Classical lossy image compression such as JPEG [1], BPG [2], and VVC Intra (VTM 12.0) [3] utilize a similar coding scheme, which strongly depends on deterministic transform features, and does not generalize well for images of diverse distributions. With the development of deep learning on computer vision, analysis transform, synthesis transform and entropy model can be replaced by convolutional neural networks [4] for lossy image compression, which has achieved better rate-distortion (RD) performances than classical methods.

In particular, an entropy model estimates the probability distribution of encoded latent representations. It plays an important role since an accurate entropy model helps to reduce bit rates in the entropy coding process. In general, there are two basic entropy modeling schemes in existing learned image compression works. One scheme is to use a hyperprior to estimate the distribution parameters [4], and another is to use a context model, which encodes the current latent symbol conditioned on previously decoded adjacent latent symbols. In order to improve the performance of entropy models, hyperprior and context model can also be combined [5–7]. For example, the context model of [6] used a multi-scale CNN to extract contextual information. In [7], hyperprior

Further author information: (Send correspondence to Ying Liu)
Tianma Shen: E-mail: tshen2@scu.edu
Ying Liu: E-mail: yliu15@scu.edu

Big Data V: Learning, Analytics, and Applications, edited by Panos P. Markopoulos,
Bing Ouyang, Vagelis Papalexakis, Proc. of SPIE Vol. 12522, 1252207
© 2023 SPIE · 0277-786X · doi: 10.1117/12.2656516

and a channel-wise auto-regressive entropy model was proposed for learned image compression. The encoded latent representation is split into slices along the channel dimension, and the decoding of the current slice is conditioned on both the hyperprior and the decoded symbols in the previous decoded slice.

Nevertheless, the aforementioned entropy models are based on CNN architectures, which are good at extracting local features only. In order to leverage long-range correlations, attention mechanisms such as spatial and channel-wise methods were proposed for learned image compression [8, 9]. However, the receptive fields of these attention methods are still limited.

In recent years, transformers [10] have been successfully utilized in many computer vision tasks such as image classification, object detection, and semantic segmentation. Compared to CNNs, transformers are better at modeling long-distance dependencies and extracting global features. Since original transformers have high computational complexity when they are applied to vision tasks, the Swin transformer [11] was proposed to improve efficiency. It is a hierarchical transformer whose representation is computed with shifted windows. The shifted windowing scheme brings greater efficiency by limiting self-attention computation to local windows while also allowing for cross-window connection. This hierarchical architecture has the flexibility to model at various scales and has linear computational complexity with respect to image size.

Most recently, transformer-based learned image compression schemes were also developed [12–15]. The first transformer-based image compression scheme is TIC [12]. Its main encoder/decoder and hyper encoder/decoder adopt a hybrid convolution-transformer structure, and its context model adopts a causal multi-head self-attention module that calculates attentions among spatially split context patches. Entroformer [13] utilizes the transformer in the hyper encoder, hyper decoder, and context model, but its analysis transform and synthesis transform adopt CNNs without any attention modules. In addition, its context model spatially splits the encoded latent representation into two slices. The decoding of the first slice is conditioned on the hyperprior only, while the decoding of the second slice is conditioned on both the hyperprior and the decoded first slice. In STF [14], the analysis and synthesis transforms adopt the Swin transformer [11], and it utilizes both hyperprior and context model for entropy modeling. In particular, its context model adopts the channel-wise auto-regressive scheme from [7]. It also has a latent residual prediction (LRP) module, which helps to further reduce the quantization error and entropy and was first proposed in [7]. From the above discussion, we conclude that both TIC [12] and Entroformer [13] utilize spatial context only, and STF [14] utilizes channel context only. Moreover, both the context model and the LRP module in STF [14] are CNN-based, which lacks the capability of learning long-range dependencies.

In this paper, we develop a novel transformer-based learned image compression architecture. Our main contributions are summarized as follows:

- We propose a new transformer-based spatial-channel auto-regressive context model for entropy modeling. Encoded latent-space features are split into spatial-channel chunks, which are entropy encoded and decoded sequentially in a channel-first order, followed by a 2D zigzag spatial order, conditioned on previously decoded feature chunks.
- We propose a transformer-based latent residual cross-attention prediction (LRCP) module, which helps to reduce quantization error.
- Experimental studies on popular image compression datasets demonstrate that our proposed method offers state-of-the-art performance visually and quantitatively.

2. THE PROPOSED METHOD

2.1 The overall architecture

Fig. 1 shows the overall architecture of our proposed learned image codec. It consists of a main encoder/decoder, a hyper encoder/decoder, and an auto-regressive entropy model. First, the main encoder compresses the input image into a latent representation $\mathbf{y}$, which is then quantized by the quantizer Q as $\widetilde{\mathbf{y}}$. In order to reduce the bits of the latent representation $\widetilde{\mathbf{y}}$, we utilize the hyperprior to estimate the distribution parameters of $\mathbf{y}$.

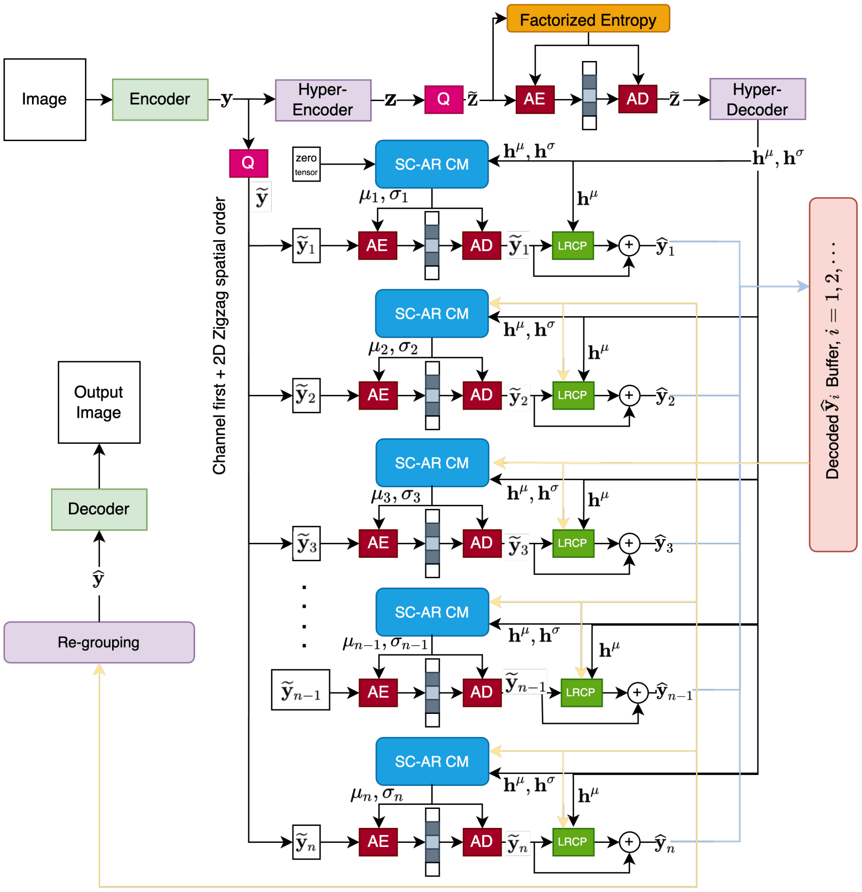
Factorized Entropy
Image
Encoder
$\mathbf{y}$
Hyper-Encoder
$\mathbf{z}$
Q
$\widetilde{\mathbf{z}}$
AE
AD
$\widetilde{\mathbf{z}}$
Hyper-Decoder
Q
zero tensor
SC-AR CM
$\mathbf{h}^{\mu}, \mathbf{h}^{\sigma}$
$\mathbf{h}^{\mu}, \mathbf{h}^{\sigma}$
$\widetilde{\mathbf{y}}$
μ_1, σ_1
$\mathbf{h}^{\mu}$
$\widetilde{\mathbf{y}}_1$
AE
AD
$\widetilde{\mathbf{y}}_1$
LRCP
$\widehat{\mathbf{y}}_1$
Channel first + 2D Zigzag spatial order
Decoded $\widehat{\mathbf{y}}_i$ Buffer, $i = 1, 2, ...$
SC-AR CM
$\mathbf{h}^{\mu}, \mathbf{h}^{\sigma}$
μ_2, σ_2
$\mathbf{h}^{\mu}$
$\widetilde{\mathbf{y}}_2$
AE
AD
$\widetilde{\mathbf{y}}_2$
LRCP
$\widehat{\mathbf{y}}_2$
Output Image
Decoder
SC-AR CM
$\mathbf{h}^{\mu}, \mathbf{h}^{\sigma}$
μ_3, σ_3
$\mathbf{h}^{\mu}$
$\widetilde{\mathbf{y}}_3$
AE
AD
$\widetilde{\mathbf{y}}_3$
LRCP
$\widehat{\mathbf{y}}_3$
$\widehat{\mathbf{y}}$
Re-grouping
SC-AR CM
$\mathbf{h}^{\mu}, \mathbf{h}^{\sigma}$
μ_{n-1}, σ_{n-1}
$\mathbf{h}^{\mu}$
$\widetilde{\mathbf{y}}_{n-1}$
AE
AD
$\widetilde{\mathbf{y}}_{n-1}$
LRCP
$\widehat{\mathbf{y}}_{n-1}$
SC-AR CM
$\mathbf{h}^{\mu}, \mathbf{h}^{\sigma}$
μ_n, σ_n
$\mathbf{h}^{\mu}$
$\widetilde{\mathbf{y}}_n$
AE
AD
$\widetilde{\mathbf{y}}_n$
LRCP
$\widehat{\mathbf{y}}_n$

Figure 1. The overall architecture of the proposed model. AE and AD are the arithmetic encoder and arithmetic decoder. Q is quantization function. SC-AR CM stands for the proposed spatial-channel auto-regressive context modal. LRCP stands for the proposed latent residual cross-attention prediction module.

The hyper encoder compresses the latent representation $\mathbf{y}$ into a hyperprior representation $\mathbf{z}$, which is quantized by the quantizer Q as $\widetilde{\mathbf{z}}$. The arithmetic encoder (AE) and the arithmetic decoder (AD) based on the distribution of the quantized representation $\widetilde{\mathbf{z}}$ convert back and forth between $\widetilde{\mathbf{z}}$ and bit steams. We apply the factorized entropy model to estimate the distribution of $\widetilde{\mathbf{z}}$. The decoded hyperprior $\mathbf{h}^{\mu}$ and $\mathbf{h}^{\sigma}$ are then utilized to estimate the distribution parameters of $\widetilde{\mathbf{y}}$ in the entropy model. The entropy-decoded latent representation $\widehat{\mathbf{y}}$ is then decompressed by the main decoder to obtain the final decoded image.

2.2 The main encoder/decoder and the hyper encoder/decoder

Fig. 2 shows the main encoder and decoder of our proposed learned image codec. The input of the main encoder is the original image of size $3 \times H \times W$, where the first dimension represents the red, green, and blue channels, H and W represent the height and width of the image, respectively. The linear embedding layer down-samples the input image by convolution layers and outputs a feature map of dimension $C \times \frac{H}{2} \times \frac{W}{2}$, where C represents the number of embedded channels. The output of the linear embedding layer is then processed by a Swin transformer block, which outputs a feature map of the same dimension $C \times \frac{H}{2} \times \frac{W}{2}$, followed by three pairs of patch merging and Swin blocks. In particular, the first patch merging block spatially splits the output feature map of the first Swin block into four patches of size $C \times \frac{H}{4} \times \frac{W}{4}$, and stacks these four patches along the channel dimension to form a feature map of size $4C \times \frac{H}{4} \times \frac{W}{4}$, followed by a linear layer to reduce the number of channels, resulting in an output feature map of size $2C \times \frac{H}{4} \times \frac{W}{4}$. The output of the first patch merging block then goes through another Swin transformer block, which outputs a feature map of the same dimension $2C \times \frac{H}{4} \times \frac{W}{4}$. This procedure is repeated twice again to output the final encoded latent representation $\mathbf{y}$ of dimension $8C \times \frac{H}{16} \times \frac{W}{16}$.

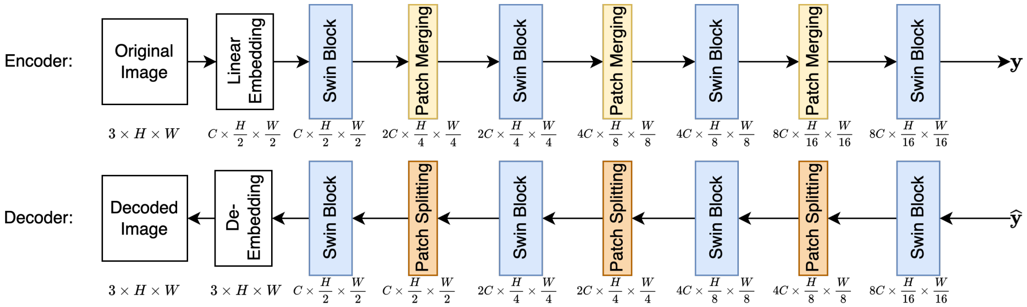
Encoder:
Original Image
Linear Embedding
Swin Block
Patch Merging
Swin Block
Patch Merging
Swin Block
Patch Merging
Swin Block
$\mathbf{y}$
$3 \times H \times W$
$C \times \frac{H}{2} \times \frac{W}{2}$
$C \times \frac{H}{2} \times \frac{W}{2}$
$2C \times \frac{H}{4} \times \frac{W}{4}$
$2C \times \frac{H}{4} \times \frac{W}{4}$
$4C \times \frac{H}{8} \times \frac{W}{8}$
$4C \times \frac{H}{8} \times \frac{W}{8}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
Decoder:
Decoded Image
De-Embedding
Swin Block
Patch Splitting
Swin Block
Patch Splitting
Swin Block
Patch Splitting
Swin Block
$\widehat{\mathbf{y}}$
$3 \times H \times W$
$3 \times H \times W$
$C \times \frac{H}{2} \times \frac{W}{2}$
$C \times \frac{H}{2} \times \frac{W}{2}$
$2C \times \frac{H}{4} \times \frac{W}{4}$
$2C \times \frac{H}{4} \times \frac{W}{4}$
$4C \times \frac{H}{8} \times \frac{W}{8}$
$4C \times \frac{H}{8} \times \frac{W}{8}$
$8C \times \frac{H}{16} \times \frac{W}{16}$

Figure 2. The architecture of the main encoder and main decoder. The dimension under each block represents the output dimension of that block. Swin Block stands for the Swin Transformer Block [11].

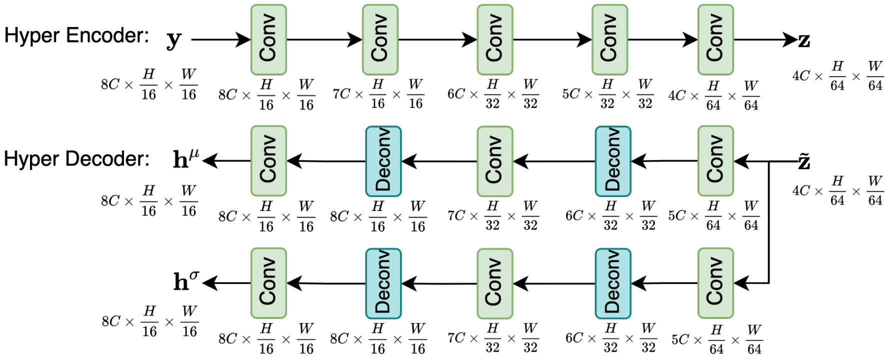
Hyper Encoder:
$\mathbf{y}$
Conv
Conv
Conv
Conv
Conv
$\mathbf{z}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
$7C \times \frac{H}{16} \times \frac{W}{16}$
$6C \times \frac{H}{32} \times \frac{W}{32}$
$5C \times \frac{H}{32} \times \frac{W}{32}$
$4C \times \frac{H}{64} \times \frac{W}{64}$
$4C \times \frac{H}{64} \times \frac{W}{64}$
Hyper Decoder:
$\mathbf{h}^{\mu}$
Conv
Deconv
Conv
Deconv
Conv
$\widetilde{\mathbf{z}}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
$7C \times \frac{H}{32} \times \frac{W}{32}$
$6C \times \frac{H}{32} \times \frac{W}{32}$
$5C \times \frac{H}{64} \times \frac{W}{64}$
$4C \times \frac{H}{64} \times \frac{W}{64}$
$\mathbf{h}^{\sigma}$
Conv
Deconv
Conv
Deconv
Conv
$8C \times \frac{H}{16} \times \frac{W}{16}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
$8C \times \frac{H}{16} \times \frac{W}{16}$
$7C \times \frac{H}{32} \times \frac{W}{32}$
$6C \times \frac{H}{32} \times \frac{W}{32}$
$5C \times \frac{H}{64} \times \frac{W}{64}$

Figure 3. The architecture of the hyper encoder and hyper decoder.The dimension under each block represents the output dimension of that block.

The input of the main decoder is the entropy-decoded latent representation $\widehat{\mathbf{y}} \in \mathbb{R}^{8C \times \frac{H}{16} \times \frac{W}{16}}$. As shown in Fig. 2, the structure of the main decoder is symmetric to that of the main encoder. It starts with one Swin block, followed by three pairs of patch splitting and Swin blocks, and a final linear de-embedding layer. In particular, the first patch splitting block utilizes a linear layer to up-sample the channel dimension of the feature map from $8C$ to $16C$ and splits the resulting feature map along the channel dimension into four patches of size $4C \times \frac{H}{16} \times \frac{W}{16}$. Afterwards, these four patches are grouped back into a feature map of size $4C \times \frac{H}{8} \times \frac{W}{8}$. This feature map is then processed by a Swin transformer block, which maintains the feature map dimension. This procedure is repeated twice again, which outputs a feature map of size $C \times \frac{H}{2} \times \frac{W}{2}$. Finally, the de-embedding layer utilizes a deconvolution layer to generate the decoded image of size $3 \times H \times W$.

Fig. 3 shows the structure of the hyper encoder and hyper decoder. The hyper encoder consists of five convolutional layers and outputs a hyperprior $\mathbf{z} \in \mathbb{R}^{4C \times \frac{H}{64} \times \frac{W}{64}}$. The hyper decoder takes the entropy decoded hyperprior $\widetilde{\mathbf{z}}$ as the input, uses two branches of five convolution/deconvolution layers to output the decoded hyperprior $\mathbf{h}^{\mu} \in \mathbb{R}^{8C \times \frac{H}{16} \times \frac{W}{16}}$ and $\mathbf{h}^{\sigma} \in \mathbb{R}^{8C \times \frac{H}{16} \times \frac{W}{16}}$.

2.3 Spatial-channel auto-regressive context model (SC-AR CM)

Our proposed auto-regressive context model splits the quantized latent representation $\widetilde{\mathbf{y}}$ into spatial-channel chunks $\widetilde{\mathbf{y}}_i$, $i = 1, 2, 3, \cdots$. These chunks are then coded sequentially in a channel-first order followed by a 2D zigzag spatial order. Fig. 4 illustrates the coding order with 3 channel slices and 3×3 spatial patches.

As shown in Fig. 1, the proposed spatial-channel auto-regressive context modal (SC-AR CM) estimates the entropy parameters μ_i and σ_i for $\widetilde{\mathbf{y}}_i$, $i = 1, 2, \cdots, n$. For the first latent chunk $\widetilde{\mathbf{y}}_1$, the SC-AR CM takes the decoded hyperprior chunks in a decoding window: $\mathbf{h}^{\mu}_j, \mathbf{h}^{\sigma}_j, j = 1, 2, \cdots, J$, where J is the window size, and an all-zero auxiliary tensor as the input, and outputs the estimated mean and standard deviation μ_1 and σ_1 of $\widetilde{\mathbf{y}}_1$, which are then taken as the input of the AE and AD. The entropy decoded $\widetilde{\mathbf{y}}_1$ is then processed by the LRCP module to output the final decoding of the first latent chunk, denoted as $\widehat{\mathbf{y}}_1$. For the second latent chunk $\widetilde{\mathbf{y}}_2$, the SC-AR CM takes the decoded hyperprior chunks $\mathbf{h}^{\mu}_j, \mathbf{h}^{\sigma}_j, j = 2, 3, \cdots, J+1$ and the decoded first latent chunk

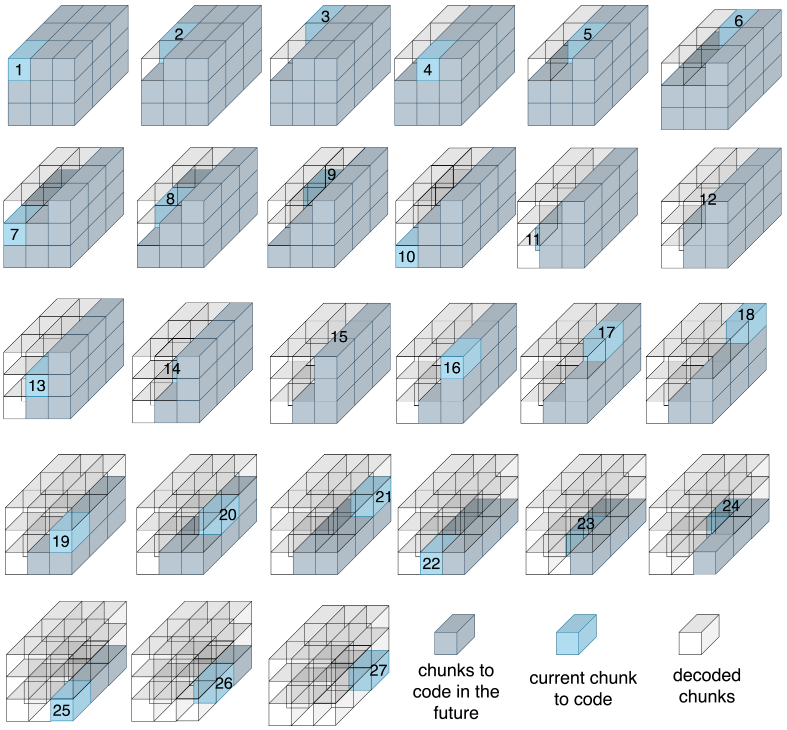

Figure 4. The coding order of the proposed spatial-channel auto-regressive context modal (SC-AR CM).

$\widehat{\mathbf{y}}_1$ as the input, and outputs the estimated mean and standard deviation μ_2 and σ_2 of $\widetilde{\mathbf{y}}_2$. For the n-th latent chunk $\widetilde{\mathbf{y}}_n$, the SC-AR CM takes the decoded hyperprior chunks $\mathbf{h}^{\mu}_j, \mathbf{h}^{\sigma}_j, j = n - J + 1, n - J + 2, \cdots, n$ and the previously decoded latent chunks $\widehat{\mathbf{y}}_{n-L}, \widehat{\mathbf{y}}_{n-L+1}, \cdots, \widehat{\mathbf{y}}_{n-1}$ as the input, where L is the window size of decoded latent chunks, and outputs the estimated mean and standard deviation μ_n and σ_n of $\widetilde{\mathbf{y}}_n$.

2.3.1 Network architecture

Fig. 5 describes the network architecture of the proposed SC-AR CM. To estimate the entropy parameters for the i-th latent chunk $\widetilde{\mathbf{y}}_i$, the inputs of the SC-AR CM are the decoded hyperprior chunks $\mathbf{h}^{\mu}_j, \mathbf{h}^{\sigma}_j, j = i, i+1, \cdots, i+J-1$ and the decoded latent chunks $\widehat{\mathbf{y}}_{i-L}, \widehat{\mathbf{y}}_{i-L+1}, \cdots, \widehat{\mathbf{y}}_{i-1}$. The context windows of size J and L constrain the amount of input to SC-AR CM, which can control the model complexity. As shown in Fig. 4, the indices of the hyperprior chuncks and the indices of the decoded latent chuncks are organized in a channel-first order, followed by a spatial 2D zigzag order. The reason is, channels have stronger correlations than spatial positions [7].

As shown in Fig. 5, the SC-AR CM has two branches. The upper branch uses 4 Swin transformer blocks to estimate the mean μ_i. The lower branch uses 2 Swin transformer blocks to estimate the standard deviation σ_i. Each of the L decoded latent chuncks and J hyperprior chuncks is spatially split into non-overlapping windows of size $M \times M$, hence the number of tokens in each window is $(L + J) \times M^2$. These tokens are the inputs of the Swin transformer blocks, and each token has a 3D position in the channel, height, and width dimensions. To encode the positional information, we propose the 3D positional embedding $\mathbf{B}$ and compute self-attention as the following

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\mathbf{Q}\mathbf{K}^T/\sqrt{d} + \mathbf{B}\right) \times \mathbf{V}, \tag{1}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the query, key and value matrices generated from the $(L + J) \times M^2$ tokens, and each element in $\mathbf{B}$ is a trainable parameter which embeds the relative 3D position of two tokens.

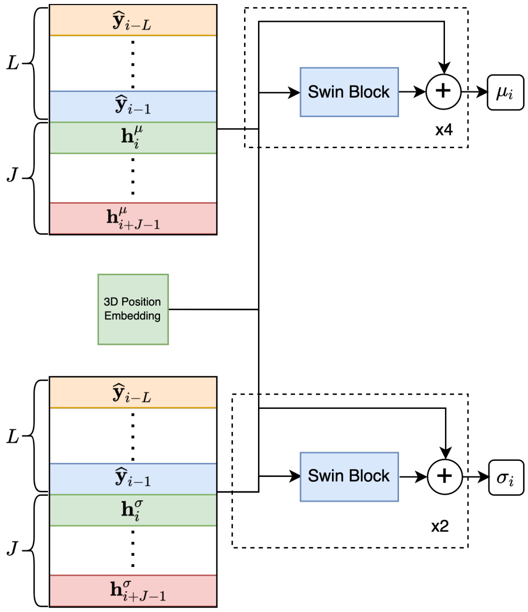
$\widehat{\mathbf{y}}_{i-L}$
$\widehat{\mathbf{y}}_{i-1}$
$\mathbf{h}_i^\mu$
$\mathbf{h}_{i+J-1}^\mu$
L
J
Swin Block
x4
μ_i
3D Position Embedding
$\widehat{\mathbf{y}}_{i-L}$
$\widehat{\mathbf{y}}_{i-1}$
$\mathbf{h}_i^\sigma$
$\mathbf{h}_{i+J-1}^\sigma$
Swin Block
x2
σ_i

Figure 5. The proposed spatial-channel auto-regressive context modal (SC-AR CM). L and J represents the context window size for $\widehat{\mathbf{y}}$ and $\mathbf{h}$, respectively.

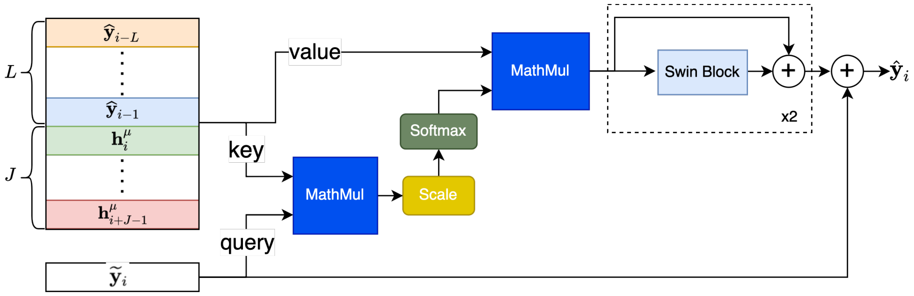

Figure 6. The proposed latent residual cross-attention prediction (LRCP) module. L and J represents the context window size for $\widehat{\mathbf{y}}$ and $\mathbf{h}$, respectively.

2.4 Latent residual cross-attention prediction (LRCP)

The quantizer Q quantizes the latent representation $\mathbf{y}$ into $\widetilde{\mathbf{y}}$, which loses some detailed information of the image. In order to recover the lost details, we propose a latent residual cross-attention prediction (LRCP) module.

Fig. 6 shows the proposed LRCP module. We integrated a cross-attention and a residual shortcut in the LRCP module, which asymmetrically combines the decoded hyperprior sequence and the decoded latent chunk sequence. The current entropy decoded latent chunk $\widetilde{\mathbf{y}}_i$ serves as a query input. The J hyperprior chunks $\mathbf{h}_j^\mu, j = i, i+1, \cdots, i+J-1$ and the L previously decoded latent chunks $\widehat{\mathbf{y}}_{i-1}, \widehat{\mathbf{y}}_{i-2}, \cdots, \widehat{\mathbf{y}}_{i-L}$ serve as the key and value inputs. In this way, LRCP can better leverage the correlation between the decoded hyperprior sequence and the decoded latent chunk sequence to compensate the quantization error.

As shown in Fig. 6, the output of the cross-attention is further processed by two Swin transformer blocks, each with a shortcut connection. Finally, the original entropy decoded $\widetilde{\mathbf{y}}_i$ is added to the output of the Swin transformer blocks by a shortcut connection to produce the final decoded $\widehat{\mathbf{y}}_i$.

2.5 Swin transformer block

A standard transformer architecture conducts global self-attention, where the relationships between a token and all other tokens are computed. Such global computation leads to quadratic complexity with respect to the number of tokens in the input sequence. To reduce the computational complexity, window-based transformer

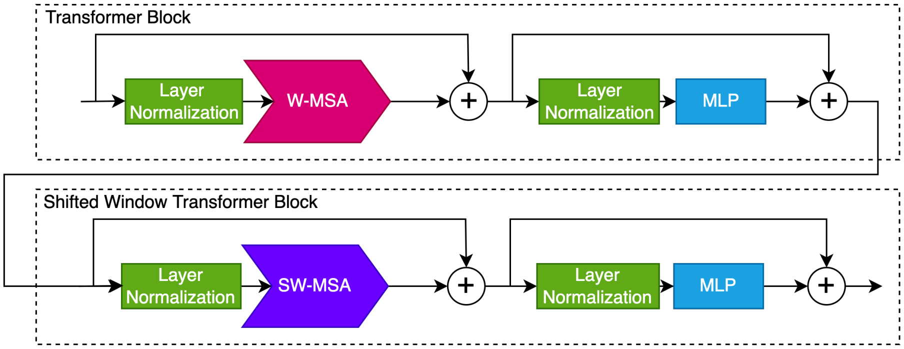

Figure 7. The architecture of the Swin Transformer Block [11].

first partitions the input of the transformer into non-overlapping windows and each window contains $M \times M$ tokens, then self-attention is computed among tokens in local windows. Further, the Swin transformer [11] was proposed to use a window shifting scheme to establish correlations among non-overlapping windows.

Fig. 7 shows the Swin transformer block [11] adopted in the main encoder/decoder, the SC-AR CM, and the LRCP module. It consists of a transformer block and a shifted window transformer block. While the transformer block adopts a multi-head self-attention module with regular windowing configuration (W-MSA), the shifted window transformer block adopts a multi-head self-attention module with shifted windowing configuration (SW-MSA).

3. EXPERIMENTAL STUDIES

3.1 Datasets

The proposed model is trained on the OpenImages dataset [16], which is known to have a diverse image distribution. We randomly chose 300,000 images from its original training set as our training set and randomly chose 10,000 images from its test set as our validation set. The resolutions of all images in these sets are higher than 546×1024. The actual training and validation images are 256×256 patches randomly cropped from these sets.

We evaluate the performance of the proposed model and existing models on the Kodak image dataset [17], Tecnick SAMPLING dataset [18], and the CLIC professional validation dataset [19]. All the models can handle images of different resolutions, and they are not limited by the training images' resolution. The Kodak dataset consists of 24 images of resolution 512×768. The Tecnick dataset consists of 40 images of resolution 1200×1200. The CLIC dataset consists of 41 images of resolution 1536×2048, which has the highest resolution and is the most complex among the three test sets.

3.2 Evaluation metrics

We quantitatively evaluate the performance of the proposed model and existing models by rate-distortion (RD) metrics. The distortion is measured by the peak signal-to-noise ratio (PSNR) and the multi-scale structural similarity (MS-SSIM) between the decoded and the ground truth images, and the bit rates are measured by the average bits per pixel (bpp) of the encoded images. Besides, we also provide qualitative experimental results to evaluate the visual quality of decoded images.

3.3 Comparison with other methods

We conduct experiments to compare the proposed model with state-of-the-art methods: STF [14], Entroformer [13] , and Coarse2Fine [20], as well as traditional image compression methods: BPG [2] and VVC Intra (VTM 12.0) [3].

The qualitative evaluation is shown in Fig. 8, which includes one sample image from each dataset. We label the bit rates (bpp), PSNR (dB), and MS-SSIM values next to the decoded images. For each sample image, we show an enlarged area to provide a close-up view of the decoding quality. Compared with STF, VTM,

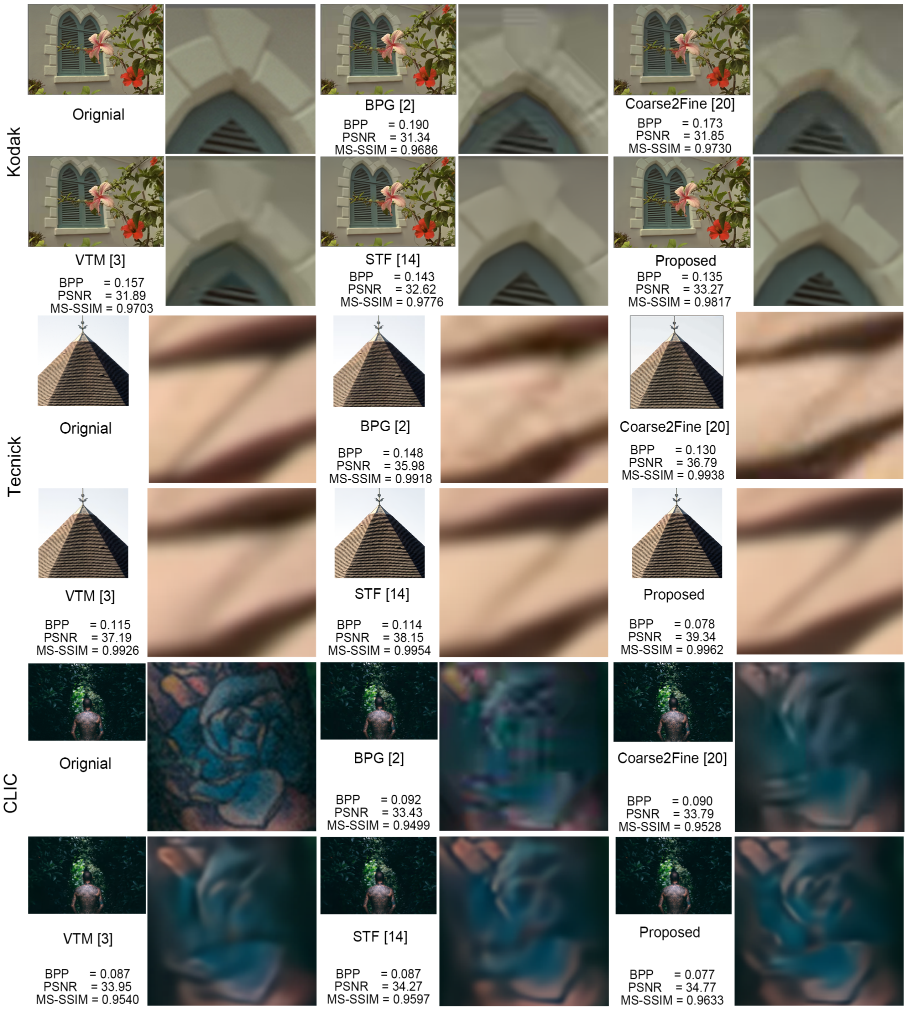

Figure 8. Sample images decoded by the proposed model, STF [14], Coarse2Fine [20], BPG [2] and VVC Intra (VTM 12.0) [3] on the Kodak dataset, Tecnick dataset, and CLIC dataset.

Coarse2Fine, and BPG, the proposed model can recover more details and textures in the decoded images, such as the edges of the triangle on the Kodak image, the line of the roof on the Tecnick image, and the flower of the man's tattoo on the CLIC image.

Meanwhile, our proposed model requires less bit rates. For the CLIC image, compared to the proposed model, the BPG, Coarse2Fine, VTM, and STF methods require 19.5%, 16.9%, 13.0%, and 13.0% extra bpp, respectively. For the Tecnick image, BPG, Coarse2Fine, VTM, and STF require 89.7%, 66.7%, 47.4%, and 46.2% extra bpp, respectively. For the Kodak image, BPG, Coarse2Fine, VTM, and STF require 40.7%, 28.1%, 16.3%, and 5.9% extra bpp, respectively.

Fig. 9 shows the PSNR and MS-SSIM curves with different bit rates on the Kodak, Tecnick and CLIC datasets, respectively. The proposed model obtains the highest PSNR and MS-SSIM scores than other methods on all datasets. Besides, at lower bit rates, the gap between the proposed model and STF is larger than the gaps at higher bit rate. In particular, on the Kodak dataset, the proposed model has increased the PSNR value of the STF model by up to 0.61 dB. For the most complex CLIC dataset, the average PSNR of the proposed model is also 0.21dB higher than that of STF.

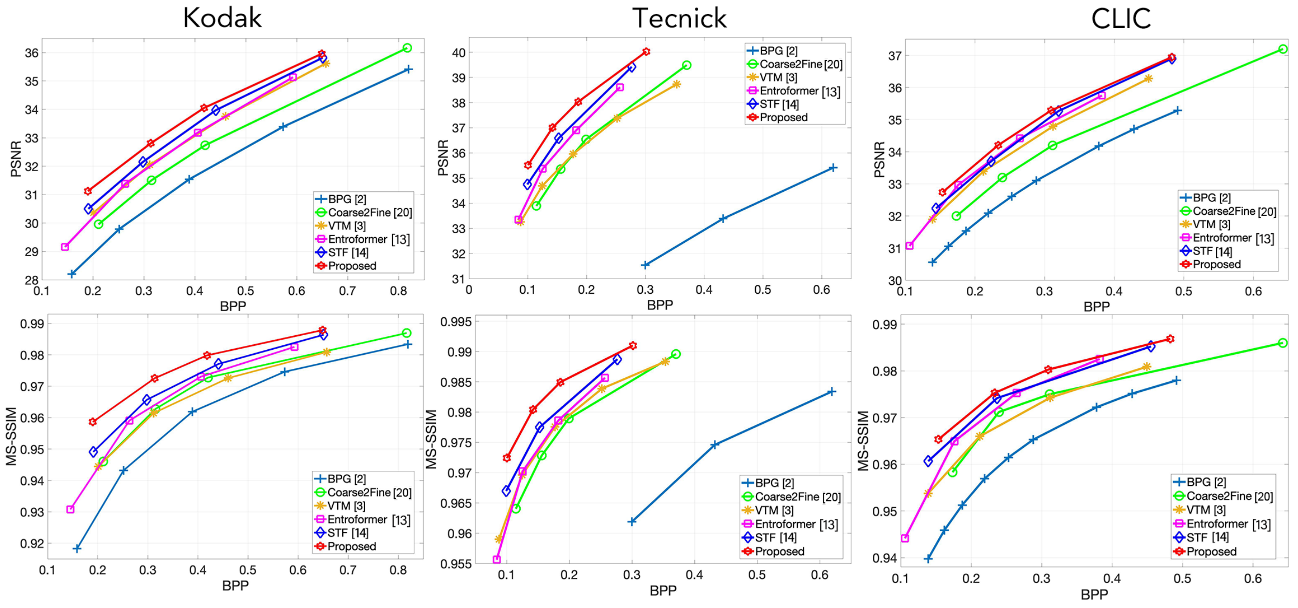

Figure 9. The PSNR and MS-SSIM of the proposed model, STF [14], Entroformer [13], Coarse2Fine [20], BPG [2] and VVC Intra (VTM 12.0) [3] on the Kodak dataset, Tecnick dataset and CLIC dataset.

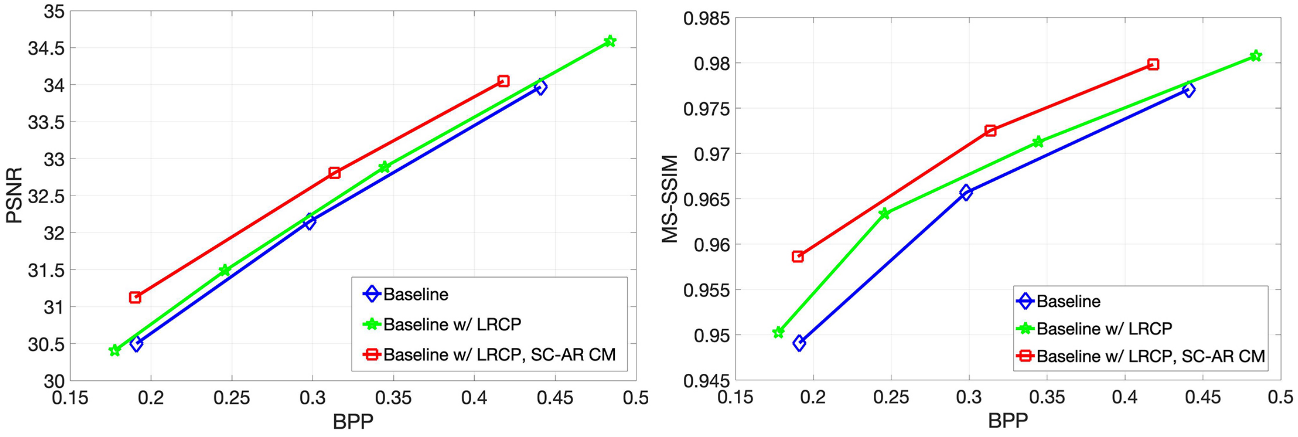

Figure 10. The PSNR (dB) and MS-SSIM of three models in the ablation study on the Kodak dataset.

3.4 Ablation study

We conducted an ablation study in Fig. 10 to demonstrate the effectiveness of different components in our proposed model. The baseline model is STF [14]. The STF model has the same main encoder/decoder, and hyper encoder/decoder as our proposed model. However, its context model adopted a CNN-based channel-wise auto-regressive entropy modeling scheme, which splits the quantized latent representation into slices along the channel direction, while our proposed model adopts a transformer-based spatial-channel auto-regressive entropy modeling scheme. STF also has a latent representation prediction (LRP) module, but it is convolution-based, while our proposed LRCP adopts a transformer-based LRP with cross-attention mechanism.

In this ablation study, we first created the model “Baseline w/ LRCP”, which replaces the convolution-based LRP of the baseline model with the transformer-based LRCP with cross-attention. The average PSNR of the “Baseline w/ LRCP” model is 0.18 dB higher than that of the baseline model on the Kodak dataset. This demonstrates that our proposed LRCP module effectively reduced the quantization error.

Further, we replaced the CNN-based channel-wise auto-regressive entropy model in the “Baseline w/ LRCP” model with the proposed transformer-based SC-AR context modal, creating the “Baseline w/ LRCP, SC-AR CM” model, which is indeed our proposed model. The average PSNR of this model is 0.33 dB higher than that of the “Baseline w/ LRCP” model on the Kodak dataset. This performance gain is higher than that between the baseline model and the “Baseline w/ LRCP” model, which indicates that the proposed SC-AR CM contributes more to the improved RD performance.

4. CONCLUSIONS

In this paper, we propose a novel learning-based image coding system using transformer structures. Our context model codes latent representations in a channel-first order, followed by a 2D zigzag spatial order. Along with transformer structures, such context model more effectively extracts contextual information for better entropy coding. Further, we propose a transformer-based latent residual cross-attention prediction (LRCP) module to reduce the quantization error. Compared to existing learned image compression approaches and traditional image compression methods, our proposed model achieved significantly better perceptual quality and RD performance. In terms of future studies, we will leverage transformer structures in the hyper encoder and decoder, and investigate the potential of the proposed learned image coding framework in other tasks such as visual coding for machines.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant ECCS-2138635 and by the NVIDIA Academic Hardware Grant Program (A100 PCIe 40GB GPU).

REFERENCES

[1] G. K. Wallace, “The jpeg still picture compression standard,” *IEEE Trans. Consumer Electronics*, vol. 38, no. 01, pp. xviii – xxxiv, Feb. 1992.

[2] F. Bellard, “BPG image format,” http://bellard.org/bpg/ (Accessed: 2022-1-18).

[3] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, “Overview of the versatile video coding (vvc) standard and its applications,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, Aug. 2021.

[4] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” in *Proc. 5th International Conference on Learning Representations*, Toulon, France, April 2017, pp. 1199–1108.

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

[6] J. Zhou, “Multi-scale and context-adaptive entropy model for image compression,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, CA, USA, Jun. 2019, pp. 799–808.

[7] D. Minnen and S. Singh, “Channel-wise autoregressive entropy models for learned image compression,” in *Proc. IEEE International Conference on Image Processing*, Abu Dhabi, United Arab Emirates, Nov. 2020, pp. 3339–3343.

[8] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learned image compression with discretized gaussian mixture likelihoods and attention modules,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Oct. 2020, pp. 7936–7945.

[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 Trans. Image Process.*, vol. 30, pp. 3179–3191, 2021.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. 9th International Conference on Learning Representations*, Virtual Event, Austria, Jun. 2021, pp. 1124–1141.

[11] 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 *Proc. 2021 IEEE/CVF International Conference on Computer Vision*, Montreal, QC, Canada, May 2021, pp. 9992–10 002.

[12] M. Lu, P. Guo, H. Shi, C. Cao, and Z. Ma, "Transformer-based image compression," in *Proc. Data Compression Conference*, Snowbird, UT, USA, Jul. 2022, p. 469.

[13] Y. Qian, X. Sun, M. Lin, Z. Tan, and R. Jin, "Entroformer: A transformer-based entropy model for learned image compression," in *Proc. The Tenth International Conference on Learning Representations*, Virtual Event, Aug. 2022, pp. 1169–1181.

[14] R. Zou, C. Song, and Z. Zhang, "The devil is in the details: Window-based attention for image compression," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, Oct. 2022, pp. 17 471–17 480.

[15] 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 *Proc. Computer Vision–ECCV*, Tel Aviv, Israel, Nov. 2022, pp. 447–463.

[16] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, and A. Kolesnikov, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, Jul. 2020.

[17] E. Kodak, "Kodak lossless true color image suite (PhotoCD PCD0992)," http://r0k.us/graphics/kodak/. (Accessed: 2022-1-18).

[18] N. Asuni and A. Giachetti, "Testimages: A large data archive for display and algorithm testing," *Journal of Graphics Tools*, vol. 17, no. 4, pp. 113–125, Oct. 2013.

[19] "Workshop and challenge on learned image compression," https://www.compression.cc/ (Accessed: 2022-1-18).

[20] Y. Hu, W. Yang, and J. Liu, "Coarse-to-fine hyper-prior modeling for learned image compression," in *Proc. The Thirty-Fourth AAAI Conference on Artificial Intelligence*, Mar. 2020, pp. 11 013–11 020.