A Low-Complexity Transformer-CNN Hybrid Model Combining Dynamic Attention for Remote Sensing Image Compression

Lili ZHANG [1], Xianjun WANG [1], Jiahui LIU [1], Qizhi FANG [1,2]

[1] School of Electronic Information Engineering, Shenyang Aerospace University, Shenyang 110136, China
[2] Liaoning General Aviation Academy, Shenyang 110136, China

{20052727, arinc2006}@sau.edu.cn, {wangxianjun, liujiahui6}@stu.sau.edu.cn

Submitted August 8, 2024 / Accepted October 22, 2024 / Online first October 31, 2024

Abstract. *Deep learning-based methods have recently made enormous progress in remote sensing image compression. However, conventional CNN is complex to adaptively capture important information from different image regions. In addition, previous transformer-based compression methods have introduced high computational complexity to the models. Remote sensing images contain rich spatial and channel information. The effective extraction of these two kinds of information for image compression remains challenging. To address these issues, we propose a new low-complexity end-to-end image compression framework combining CNN and transformer. This framework includes two critical modules: the Dynamic Attention Model (DAM) and the Hyper-Prior Hybrid Attention Model (HPHAM). By employing dynamic convolution as the core part of the DAM, the DAM can dynamically adjust the attention weights according to the image content. HPHAM effectively integrates non-local and channel information of latent representations through the parallel running of Gated Channel Attention (GCA) and multi-head self-attention. Experiments demonstrate that the proposed approach outperforms existing mainstream deep-learning image compression approaches and conventional image compression methods, achieving optimal rate-distortion performance on three datasets. Code is available at https://github.com/jiahuiLiu11/LTCHM.*

Keywords

Remote sensing image compression, dynamic convolution, attention mechanism, gating mechanism

1. Introduction

In the current era of information explosion, remote sensing technology has become an indispensable tool in many fields, including earth sciences, environmental monitoring, and agricultural management. With the development of remote sensing technology and the continuous improvement of satellite resolution, the data volume of remote sensing images is also increasing dramatically, which brings significant challenges in data storage, transmission, and processing. Therefore, efficient remote sensing image compression techniques are essential. In the past decades, conventional image compression methods, such as JPEG [1], JPEG2000 [2], HEVC/H.265 [3], and CCSDS [4], have achieved good rate-distortion performance for image compression. However, these image compression methods rely heavily on hand-crafted codec modules and have complex interdependencies. It is challenging to integrate the entire encoding and decoding system. The encoding and decoding process needs to be implemented block by block sequentially in a block-based hybrid codec, which brings block effect and ringing effect to the reconstructed image [5]. Thus, they frequently struggle to achieve satisfactory results when dealing with remote-sensing images of higher complexity. As deep learning technology advances quickly, its powerful potential offers fresh approaches and ideas for resolving the issue of remote sensing image compression.

Remote sensing image compression technology aims to effectively reduce the data volume of remote sensing images while preserving critical information. Currently, traditional compression methods remain the mainstream approach for processing remote sensing images. For a hybrid lossless compression technique, Afjal et al. [6] suggested a band reordering scheme for segmented subgroup datasets of the original remote sensing image dataset. Hong et al. [7] explored the effectiveness of its application in remote sensing image compression using discrete cosine transform. For compressing remote sensing images with high spatial resolution, Zhang et al. [8] proposed a fast discrete wavelet transform (DWT) based on orientation prediction. Xiang et al. [9] proposed a band selection and slant-haar orthogonal transform-based hyperspectral image compression technique for remote sensing. Shi et al. [10] proposed an adaptive scanning remote sensing image compression method based on the human visual system.

DOI: 10.13164/re.2024.0642

Deep learning, especially CNN and autoencoders, has shown excellent performance in image recognition, classification, and generation tasks. In recent studies, most deep learning-based image compression methods usually use end-to-end training strategies. End-to-end learning image compression [11–13] allows for optimizing the entire framework. This approach allows for the automatic optimization of feature extraction and information encoding during the encoding process. Additionally, it adjusts the model structure and parameters to meet specific compression needs, resulting in improved compression performance. In terms of Peak Signal-to-Noise Ratio (PSNR) and Multi-scale Structural Similarity (MS-SSIM) [14], some recent work on learning image compression [15–21] performs better than traditional image compression algorithms. This shows that deep learning-based image compression techniques have great potential in the future.

As deep learning technology has advanced over the past few years, an increasing number of research projects have used deep learning in the application of remote sensing image compression. Fu et al. [22] proposed a hybrid hyperprior network based on transformers and CNN. The network can explore local and non-local redundancies to improve entropy estimation accuracy. To effectively adjust the feature distribution and reduce the information dependence in Synthetic Aperture Radar (SAR) image compression, Zhang et al. [23] presented an end-to-end trainable model using a discrete Gaussian adaptive entropy model. In addition, generalized subtractive normalization is applied to minimize the remaining redundancy and reduce the statistical properties of SAR images. Zhang et al. [24] proposed a region-of-interest compression algorithm based on a deep learning autoencoder framework to improve image reconstruction performance and reduce distortion in the region of interest. Zhang et al. [25] extended the hyperprior with a global stripe self-attention mechanism to capture global, local, and channel dependencies. It enables global correlation and hierarchical modeling of latent vectors. A multi-scale depth-wise convolution-based attention module was introduced to boost the feature extraction capabilities of the encoder and decoder. This module increases the receptive field and nonlinear transformation capacity, which retains more valuable information for compression.

The majority of deep learning-based image compression methods [11, 19, 26–28] are designed using Variational Auto-Encoder (VAE) [29] and CNN. Typically, VAE-based image compression consists of three main steps: transformation, quantization, and entropy encoding. The transformation process involves converting the original image into the potential coding space to create a latent representation. Deep learning-based image compression typically applies uniform quantization during the quantization process. This method divides the input signal's value range into equal-width intervals, called "quantization steps" or "quantization intervals," and rounds each input value to the nearest discrete value. Since the derivative of uniform quantization is nearly zero at most points, gradient descent becomes ineffective. Previous research [11] proposes using additive uniform noise during the training phase as a proxy for quantization to allow optimization through stochastic gradient descent. In contrast, uniform quantization is used during the inference stage. Entropy encoding uses an entropy model to estimate the entropy of the latent representation. Creating an efficient and accurate entropy model is critical to improving image compression performance. Thus, Ballé et al. [12] introduced a VAE-based image compression model, leveraging a hyperprior structure to capture spatial dependencies in latent representations. Additionally, side information is used to estimate the variance of the parameter distribution. To improve the utilization of the probability distribution of the potential representation in the compression model, Minnen et al. [13] proposed the merger of the autoregressive model and the hyper-prior. Numerous image compression models have been proposed based on this. With the rise of visual transformers [30], [31], researchers have begun to explore their application in image compression. Qian et al. [20] proposed a new entropy model based on transformers, effectively capturing long-range dependencies in probability distribution estimation. Koyuncu et al. [21] suggested Transformer-based contextual modeling motivated by the adaptive features of the Transformer. Khoshkhahtinat et al. [32] proposed transformer-based nonlinear transformations to overcome the limitations of CNN. However, remote sensing images typically have richer information and higher resolution than regular images. Therefore, it is necessary to collect global and local information to enhance the compression performance of remote sensing images. Although CNN is constrained by its receptive fields and lacks the ability to model long-term dependencies, it has solid local feature extraction capability. Transformers use multi-head self-attention to capture long-range dependencies and non-local features effectively. Both methods have distinct advantages, but merging these strengths into a unified approach remains a significant challenge. A parallel Transformer-CNN hybrid module (TCM) that combines the non-local modeling capabilities of the Transformer with the local modeling capabilities of CNN was proposed by Liu et al. [33]. However, the computational complexity of the entire compression model is boosted by this design. The main reason is the significant computational complexity of the multi-head self-attention mechanism in Transformers, coupled with the entire compression framework being constructed solely from TCM.

Over the past few years, attention mechanisms have been widely utilized for computer vision tasks. It mimics human information processing by efficiently concentrating on important information while filtering out irrelevant details. Image compression architectures have incorporated a variety of attention models to enhance the rate-distortion performance of image compression [17, 19, 26, 33]. Zou et al. [17] proposed a simple and efficient window-based attention model. Chen et al. [19] proposed a non-local attention mechanism for generating implicit masks to weight adaptive bit allocation features.

Cheng et al. [26] contended that the non-local attention mechanism is extremely time-consuming during training. Consequently, they propose a simple attention mechanism by removing the non-local operation block. Liu et al. [33] suggested a parameter-effective swin-transformer-based attention model to reduce the complexity of the model by putting the attention model into an entropy model. Window-based local attention models offer low computational complexity but lack inter-window information exchange, hindering non-local information extraction. Although computationally expensive, non-local attention mechanisms capture broader contextual information. Simple attention mechanisms reduce training time but are limited in feature extraction capabilities. Moving the Swin-Transformer-based attention model within the entropy model can reduce the overall computational complexity of the compression model. However, the attention model still faces high computational complexity.

To address the above issues, we introduce a new image compression framework called the Low-complexity Transformer-CNN Hybrid Model (LTCHM). The contribution of this paper can be summarized as follows:

- In order to reduce the computational complexity of the model, in this paper, we use the CNN residual block and Dynamic Attention Model (DAM) to form the primary codec and use the swin-transformer-based Hyper-Priori Hybrid Attention Model (HPHAM) in the hyper-priori structure. The rate-distortion performance is improved by effectively combining local, non-local, and channel information.

- To strike an optimal balance between computational complexity and rate-distortion performance in attention models, we provide a Dynamic Attention Model (DAM) that utilizes dynamic convolution. DAM not only significantly improves the rate-distortion performance of the model but also further reduces the computational effort. By adaptive adjusting the weights of the input features, DAM can flexibly focus on significant regions in the image, thus capturing local and global information more effectively.

- Our Hyper-Prior Hybrid Attention Model (HPHAM) employs Window-based Multi-head Self-Attention (W-MSA) and Shifted Window-based Multi-head Self-Attention (SW-MSA) alongside gated channel attention in parallel, respectively. This integration effectively merges channel and non-local information, improving the rate-distortion performance of the compression model.

- We introduce a novel feed-forward network called the Multi-Scale and Multi-Branch Feed-Forward Network (MSB-FFN). This network incorporates a hierarchical structural design and utilizes multi-branch parallel processing. It enhances the ability of the network to represent multi-scale features at a more detailed level, resulting in the effective capture of multi-scale information.

- Experimental results demonstrate that our approach outperforms traditional and deep learning-based image compression methods on three distinct remote sensing datasets (i.e., DOTA, UC-Merced, and China Gaofen satellite datasets).

The structure of this paper is as follows: Section 2 presents the proposed low-complexity Transformer-CNN hybrid model for image compression in detail. Section 3 discusses the experimental setup and datasets used, along with an analysis and comparison of the results to evaluate the effectiveness of the proposed method. Finally, Section 4 concludes the paper and suggests directions for future research.

2. Proposed Method

2.1 Overall Framework

LTCHM is built based on the Gaussian hybrid entropy model proposed by Cheng et al. [14]. As shown in Fig. 1, the structure contains the primary encoder, the primary decoder, the hyper-priori encoder, the hyper-priori decoder, and the entropy model. A DAM and a residual block constitute the primary encoder and decoder. By dynamically adjusting attention levels across different regions, the DAM strengthens the capacity of the network to extract relevant information. The hyper-prior encoder and decoder network are composed of HPHAM and down-sampling modules. To enhance compression performance, HPHAM extracts global and channel information. Assumed an original image with dimension $\mathbf{x} \in 3 \times H \times W$, the master encoder converts it into the latent representation $\mathbf{y} \in C \times \frac{H}{16} \times \frac{W}{16}$. To create $\hat{\mathbf{y}} \in C \times \frac{H}{16} \times \frac{W}{16}$, a discrete expression of the potential representation $\mathbf{y}$, $\mathbf{y}$ is quantized. The process of quantization unavoidably introduces errors, which distorts the reconstructed image. As such, Ballé et al. [11] suggested adding uniform noise $\mathcal{U}(-0.5, 0.5)$ approximate quantization during the training phase to produce noisy encodings. At inference, $\mathbf{y}$ is rounded using the round function to obtain the latent representation and correct quantization errors. Finally, $\hat{\mathbf{y}}$ is input into the primary decoder to generate the reconstructed image $\hat{\mathbf{x}} \in 3 \times H \times W$.

An entropy model is frequently used to estimate the parameterized distribution of $\hat{\mathbf{y}}$, which is essential for rate-distortion performance. It is necessary to model the conditional probability distribution $p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}}(\hat{\mathbf{y}}|\hat{\mathbf{z}})$ after decoding $\hat{\mathbf{z}}$. Therefore, we model $p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}}(\hat{\mathbf{y}}|\hat{\mathbf{z}})$ using the Gaussian mixture entropy model proposed by Cheng et al. [26], i.e.:

$$p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}}(\hat{\mathbf{y}}|\hat{\mathbf{z}}) \sim \sum_{k=1}^{K} \boldsymbol{\omega}^{(k)} \mathcal{N}\left(\boldsymbol{\mu}^{(k)}, \boldsymbol{\sigma}^{2(k)}\right). \tag{1}$$

This can be further expressed as [26]:

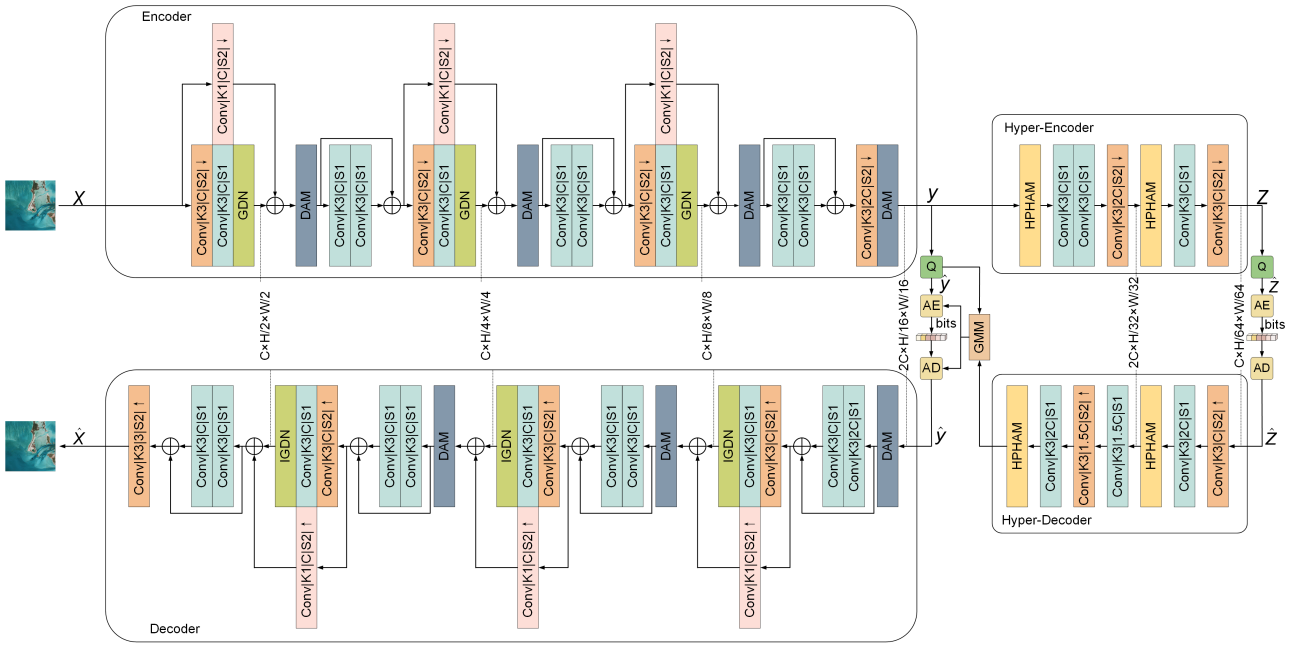

Fig. 1. The overall framework of LTCHM. GMM denotes the Gaussian Mixed entropy Model. DAM denotes Dynamic Attention Module. HPHAM denotes the Hyper-Prior Hybrid Attention Model. K denotes convolutional kernel size, $K3$ denotes a convolutional kernel size of 3, $K1$ denotes a convolutional kernel size of 1. C denotes the number of channels, $C = 192$ when $\lambda = \{128, 256, 512\}$ and $C = 256$ when $\lambda = \{1024, 2048, 4096\}$. S denotes convolutional step size, $S1$ denotes a step size of 1, $S2$ denotes a step size of 1. $\downarrow$ denotes down-sampling, and $\uparrow$ denotes up-sampling.

$$
\begin{aligned}
p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}}(\hat{\mathbf{y}}|\hat{\mathbf{z}}) &= \prod_i p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}}\left(\hat{y}_i|\hat{\mathbf{z}}\right), \\
p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}}\left(\hat{y}_i|\hat{\mathbf{z}}\right) &= \left(\chi_i * \mathcal{U}\left(-0.5, 0.5\right)\right)\left(\hat{y}_i\right), \\
\chi_i &= \left(\sum_{k=1}^{K} \omega_i^{(k)} \mathcal{N}\left(\mu_i^{(k)}, \sigma_i^{2(k)}\right)\right)
\end{aligned} \tag{2}
$$

where χ_i denotes i-th feature of the potential representation and i represents the position of the feature map in the mapping. The symbol k signifies the exponent of the mixture. Each mixture Gaussian distribution contains three parameters, i.e., weight $\omega_i^{(k)}$, $\mu_i^{(k)}$, and $\sigma_i^{2(k)}$ for each element.

The problem is regarded as a rate-distortion optimization problem based on Lagrange multipliers, which is significant for training the whole compression model. The loss function is defined as [26]:

$$
\begin{aligned}
\mathcal{L} &= \mathcal{R}\left(\hat{\mathbf{y}}\right) + \mathcal{R}\left(\hat{\mathbf{z}}\right) + \lambda \cdot \mathcal{D}\left(\mathbf{x}, \hat{\mathbf{x}}\right) \\
&= \mathbb{E}\left[-\log_2\left(p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}}\left(\hat{\mathbf{y}}|\hat{\mathbf{z}}\right)\right)\right] + \mathbb{E}\left[-\log_2\left(p_{\hat{\mathbf{z}}|\boldsymbol{\psi}}\left(\hat{\mathbf{z}}|\boldsymbol{\psi}\right)\right)\right] \\
&+ \lambda \cdot \mathcal{D}\left(\mathbf{x}, \hat{\mathbf{x}}\right)
\end{aligned} \tag{3}
$$

where λ is used to control the rate-distortion trade-off. Different values of λ correspond to different bit rates. $\mathcal{D}\left(\mathbf{x}, \hat{\mathbf{x}}\right)$ denotes the distortion between the original image and the reconstructed image, which can be computed using the MSE or the MS-SSIM. $\mathcal{R}\left(\hat{\mathbf{y}}\right)$ and $\mathcal{R}\left(\hat{\mathbf{z}}\right)$ signify the bit rates of potential $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$, respectively. Since $\hat{\mathbf{z}}$ has no prior, it is encoded using the factorized density model $\boldsymbol{\psi}$, i.e. [26]:

$$
p_{\hat{\mathbf{z}}|\boldsymbol{\psi}}\left(\hat{\mathbf{z}}|\boldsymbol{\psi}\right) = \prod_i \left(p_{z_i|\boldsymbol{\psi}}\left(\boldsymbol{\psi}\right) * \mathcal{U}\left(-0.5, 0.5\right)\right)\left(\hat{z}_i\right) \tag{4}
$$

where z_i represents the i-th element of $\mathbf{z}$, i denotes the position of each element.

2.2 Dynamic Attention Model

The Dynamic Attention Model (DAM) incorporates the Input-dependent Depth-wise Convolution (IDConv) as a core component, as proposed by Lou et al. [34]. The structure of IDConv is shown in Fig. 2(a). IDConv takes the feature map $X \in \mathbb{R}^{C\times H\times W}$ as input and uses adaptive average pooling to reduce the spatial dimensions from $H \times W$ to $K \times K$. Then, two 1×1 convolution layers are employed to generate and refine the feature map precisely. The first 1×1 convolution layers produce feature map $F' \in \mathbb{R}^{\frac{C}{r}\times K\times K}$, which is subsequently fed into the second 1×1 convolution layer to obtain the output feature map $F'' \in \mathbb{R}^{(G\times C)\times K\times K}$, where G denotes the number of feature map groupings. The dimension of feature map F'' is resized to $\mathbb{R}^{G\times C\times K\times K}$, optimized with the softmax function, obtaining the output feature map weight $F \in \mathbb{R}^{G\times C\times K\times K}$. This weight is multiplied element-by-element with a set of learnable parameters $L \in \mathbb{R}^{G\times C\times K\times K}$ and summed over dimension G to integrate relevant information, which forms the input-dependent depth-wise convolution kernel $D \in \mathbb{R}^{C\times K\times K}$. The dynamic convolution kernel is used to convolve the input feature X, producing the final output Out_{IDConv}, represented as [34]:

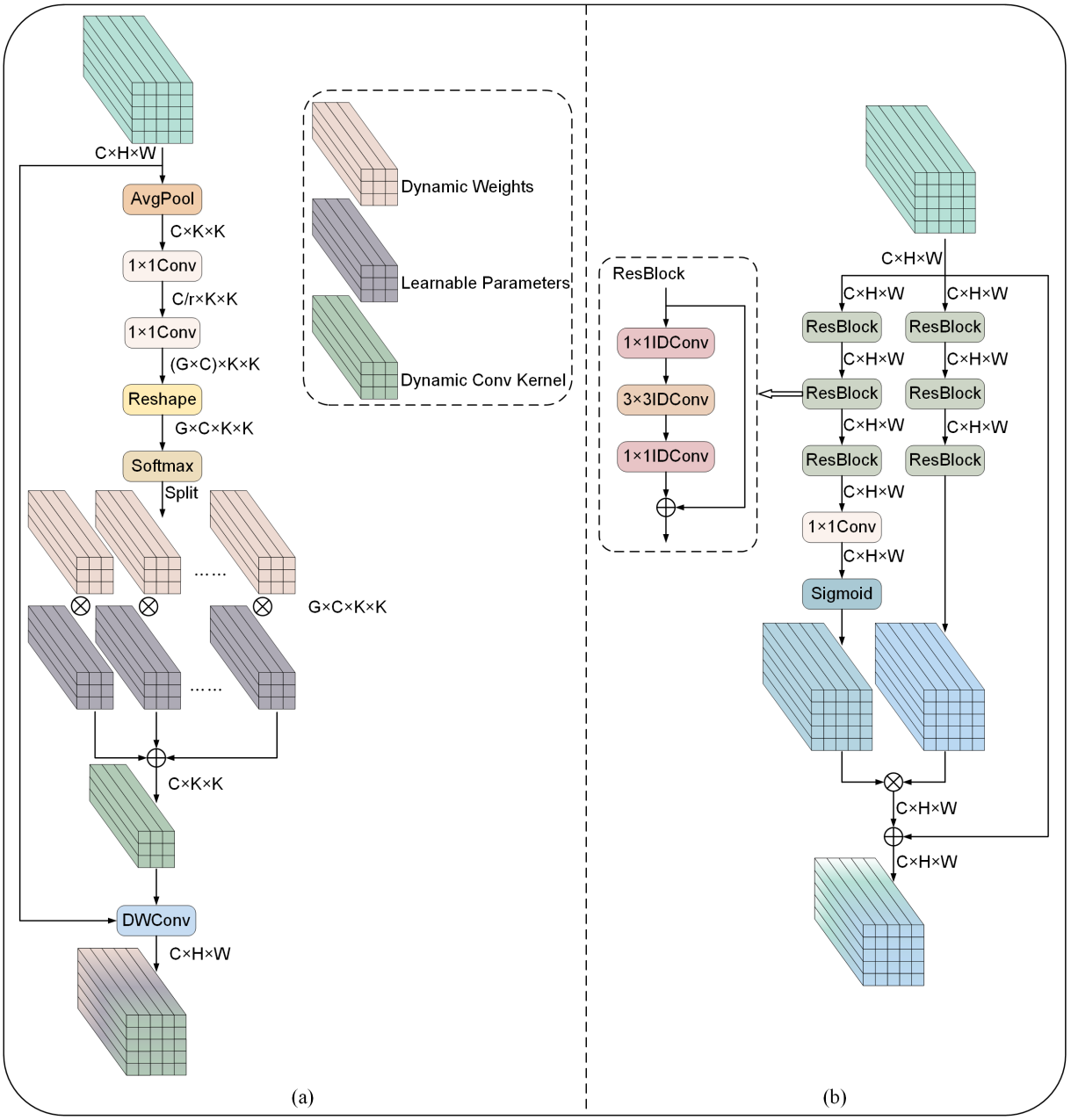

Fig. 2. (a) Input-dependent depth-wise convolution. C denotes the number of channels, $C = 192$ when $\lambda = \{128, 256, 512\}$ and $C = 256$ when $\lambda = \{1024, 2048, 4096\}$. r denotes the channel attenuation ratio set to 2 in the experiments. K denotes the convolutional kernel size set to 1 and 3 in the experiments. G denotes the number of groups set to 2 in the experiments. 1×1Conv means that the convolution kernel size is 1, the step size is set to 1, and the number of channels is set to $\frac{C}{r}$ and $(G \times C)$, respectively. (b) Dynamic attention module. 1×1Conv means that the convolution kernel size is 1, the step size is 1, and the number of channels is set to C.

$$
\begin{aligned}
F' &= \mathrm{Conv}_{1\times1}\left(\mathrm{AdaptiveAvgPool}\left(X\right)\right), \\
F'' &= \mathrm{Conv}_{1\times1}\left(F'\right), \\
F &= \mathrm{Softmax}\left(\mathrm{Reshape}\left(F''\right)\right), \\
D &= \sum_{i=0}^{G} L_i F_i, \\
Out_{\mathrm{IDConv}} &= D\left(X\right).
\end{aligned} \tag{5}
$$

According to research in [5, 17, 19, 26, 33], the attention mechanism can enhance the rate-distortion performance of image compression. To further enhance the rate-distortion performance of remote sensing image compression, we design DAM based on IDConv. Because dynamic convolution can produce distinct weight parameters based on varying inputs, it can extract local and non-local information more effectively. As illustrated in Fig. 2(b), the DAM is created by building residual blocks with IDConv and cascading the residual block.

Dynamic convolution generates convolution kernels dynamically during network forward propagation. This operation enables feature extraction at each layer to adaptively adjust according to the attributes of the input feature map. Unlike the conventional fixed convolutional kernel method, this approach allows the network to adjust its parameters dynamically. Dynamic attention mechanisms highlight significant information by assigning different weights to various parts of the input feature map. Contrary to the fixed attention mechanism, dynamic attention may adapt the attention allocation in real-time based on the variations in the content of the feature map. It further enhances the ability of the model to identify and handle significant characteristics.

By integrating IDConv as the core component of the attention module, we can construct a highly effective adaptive attention model. In this model, dynamic convolution generates convolution kernels dynamically based on input data characteristics, while the dynamic attention mechanism assigns varying weights according to feature importance. This dual dynamic adjustment mechanism allows the model to handle complex data more accurately and efficiently. During forward propagation, convolution kernels are generated in real-time based on input data analysis, followed by rapid parameter generation. These convolutional kernels extract features, and attention weights are dynamically assigned based on these features. The dynamic attention mechanism reweights features according to their importance, enhancing critical features and suppressing less important ones.

2.3 Gated Channel Attention

Narayanan et al. [35] proposed the Squeeze Aggregated Excitation (SaE) module. This module uses multi-branch fully connected layers to process compressed features in parallel, enhancing the ability of the model to capture global channel information by aggregating the outputs of these branches. Increasing the number of branches improves the feature representation capability of the network without significantly increasing model parameters. Inspired by this, we propose a Gated Channel Attention (GCA) mechanism to aid compression by extracting channel information, as shown in Fig. 3. First, global average pooling compresses the input tensor to aggregate channel information. The pooled output is divided into four tensors along the channel dimension. Next, four parallel 1×1 convolutional layers process each tensor, with a gating mechanism designed after each convolution. This gating mechanism dynamically adjusts the channel weights based on their importance in enhancing the feature representation. Finally, the four tensors are concatenated and passed through another 1×1 convolutional layer to re-aggregate the channel features. The output of the 1×1 convolutional layer is channel-by-channel multiplied by the original input tensor to form the final output.

2.4 Multi-Scale and Multi-Branch Feed-Forward Network

The role of the feedforward network is to integrate and map global dependencies between different feature representations. Multi-scale features are crucial for computer vision tasks as they better capture objects of varying sizes, contextual information, and non-local features. Building on the work of [34], [36], we propose a Multi-Scale and Multi-Branch Feed-Forward Network (MSB-FFN), as shown in Fig. 4. Following the 1×1 convolution layer, the feature map is evenly divided into four parts, each processed by parallel depth-wise separable convolutions of different scales. Each convolution handles a quarter of the channels to capture

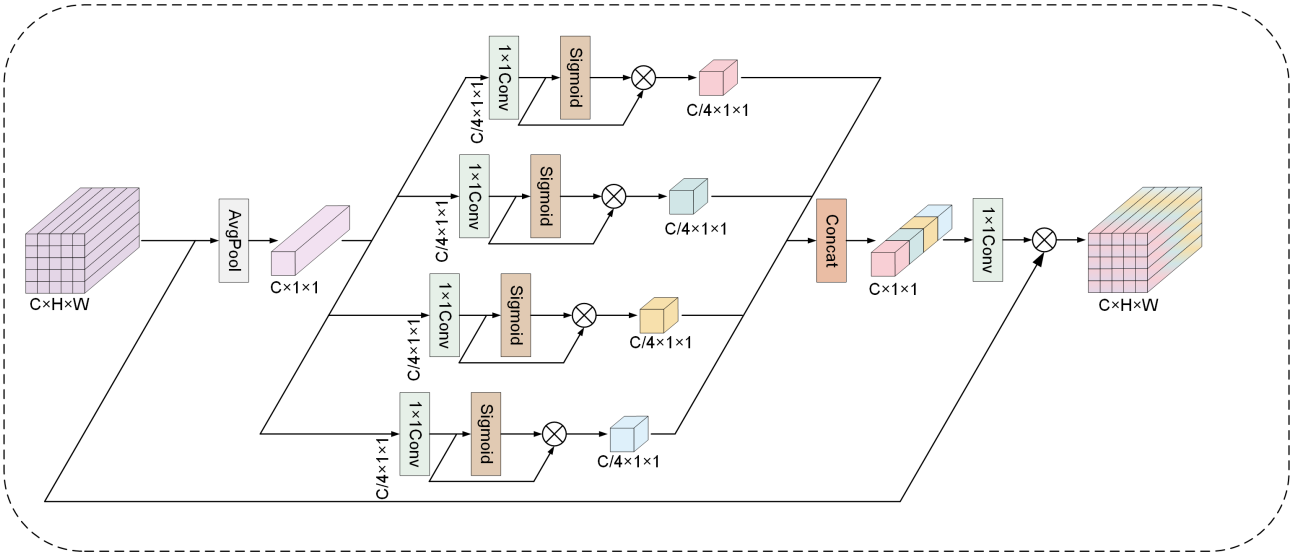

Fig. 3. Gated Channel Attention. C denotes the number of channels, $C = 192$ when $\lambda = \{128, 256, 512\}$ and $C = 256$ when $\lambda = \{1024, 2048, 4096\}$. 1×1Conv means that the convolution kernel size is 1, the step size is 1, and the number of channels is set to C.

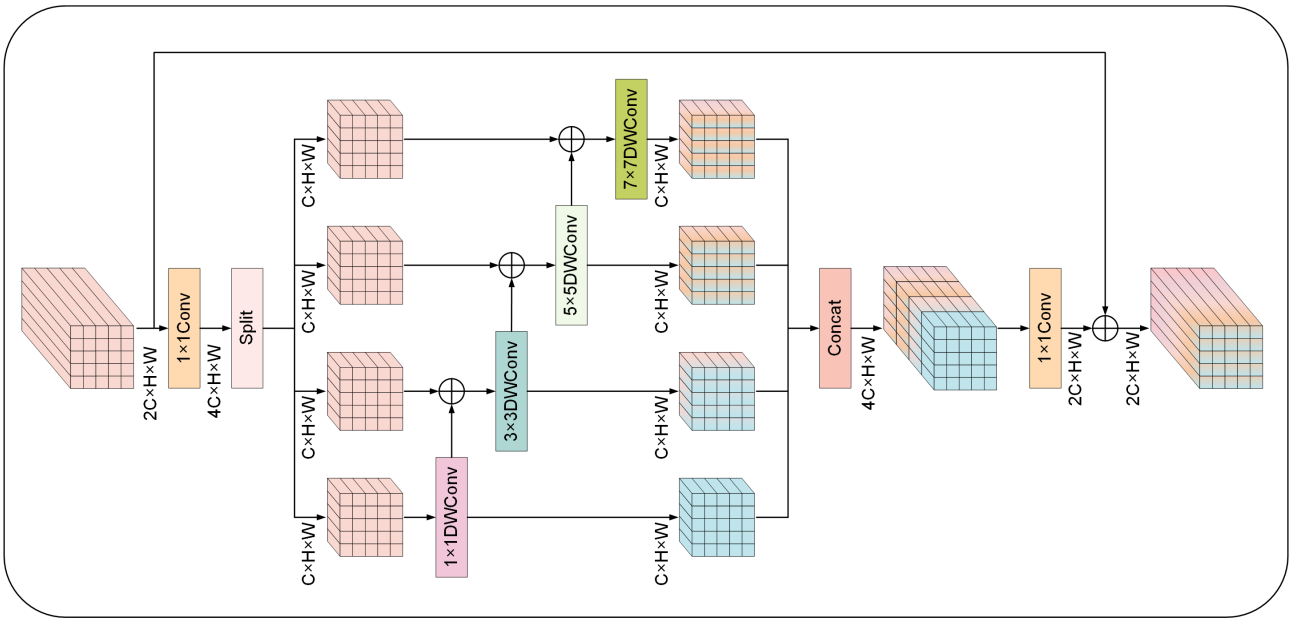

Fig. 4. Multi-Scale and Multi-Branch Feed-Forward Network. C denotes the number of channels, $C = 192$ when $\lambda = \{128, 256, 512\}$ and $C = 256$ when $\lambda = \{1024, 2048, 4096\}$. 1×1DWConv denotes a depth separable convolution with convolution kernel size 1. 3×3DWConv denotes a depth separable convolution with convolution kernel size 3. 5×5DWConv denotes a depth separable convolution with convolution kernel size 5. 7×7DWConv denotes a depth separable convolution with convolution kernel size 5. The step size of the four depth separable convolutions is 1, and the number of channels is C.

multi-scale information. Following the convolutional layer, the first three sets of features are split into two branches. One of the branches will continue to propagate forward, while the other branch will be transferred to the convolutional layer along with the information flow route of the latter set of input features. The MSB-FFN leverages hierarchical residual connections and multi-branch parallel processing to boost the multi-scale feature capture and representation capabilities of the model. This method broadens the receptive field of each layer and enhances the multi-scale feature representation at a more detailed level.

2.5 Hyper-Prior Hybrid Attention Model

The vision transformer-based image compression methods, as proposed by references [20] and [33], have achieved remarkable success in image compression. By relying on the multi-head self-attention mechanism of transformers, these methods capture global information from images and establish efficient long-range dependencies. According to references [21] and [32], channel information can significantly boost the performance of compression models. Building on this insight, we have developed a Hyper-Prior Hybrid Attention Model (HPHAM), which merges the multi-head self-attention mechanism with Gated Channel Attention (GCA). As shown in Fig. 5, the HPHAM proceeds through two stages, each following a similar procedure. In the first stage, a window-based multi-head self-attention (W-MSA) module is in conjunction with GCA. The W-MSA extracts local information, which is combined with the channel information obtained by GCA. During the second stage, it combines a shifted window-based multi-head self-attention module (SW-MSA) with GCA. Using SW-MSA, non-local information is retrieved and then integrated with channel information. Assume the dimension of the input tensor X is $\mathbb{R}^{2C\times H\times W}$. The tensor X is initially fed into a 1×1 convolutional layer. Subsequently, the output tensor of the 1×1 convolutional layer is split equally along the channel dimensions into two tensors, $X_{\text{cha}} \in \mathbb{R}^{C\times H\times W}$ and $X_{\text{trans}} \in \mathbb{R}^{C\times H\times W}$. Following that, the tensor $X_{\text{trans}} \in \mathbb{R}^{C\times H\times W}$ is promptly fed into the W-MSA to obtain $X'_{\text{trans}} \in \mathbb{R}^{C\times H\times W}$, while the tensor $X_{\text{cha}} \in \mathbb{R}^{C\times H\times W}$ is fed into the GCA to obtain $X'_{\text{cha}} \in \mathbb{R}^{C\times H\times W}$. Subsequently, the tensors X'_{trans} and X'_{cha} are concatenated and inputted into the 1×1 convolutional layer to enhance the integration of channel features and local features. Finally, the output X_{mid} is then obtained by creating a jump link between the 1×1 convolutional layer and the input feature X. X_{mid} is fed into a Multi-Scale and Multi-Branch Feed-Forward Network (MSB-FFN) to obtain the output X_{out} of the first stage. The second stage utilizes the output of the first stage as its input, and the remainder of the procedure is analogous to the first stage. Consequently, the second stage will not be explicitly described. The first stage is represented by the equation:

$$\begin{aligned} X_{\text{cha}}, X_{\text{trans}} &= \text{Split}\left(\text{Conv}_{1\times1}\left(\text{LayerNorm}\left(X\right)\right)\right), \\ X'_{\text{cha}} &= \text{GCA}\left(X_{\text{cha}}\right), \\ X'_{\text{trans}} &= \text{W-MSA}\left(X_{\text{trans}}\right), \\ X_{\text{mid}} &= \text{Conv}_{1\times1}\left(\text{Concat}\left(X'_{\text{cha}}, X'_{\text{trans}}\right)\right) + X, \\ X_{\text{out}} &= \text{MSB-FFN}\left(\text{LayerNorm}\left(X_{\text{mid}}\right)\right) + X_{\text{mid}}. \end{aligned} \tag{6}$$

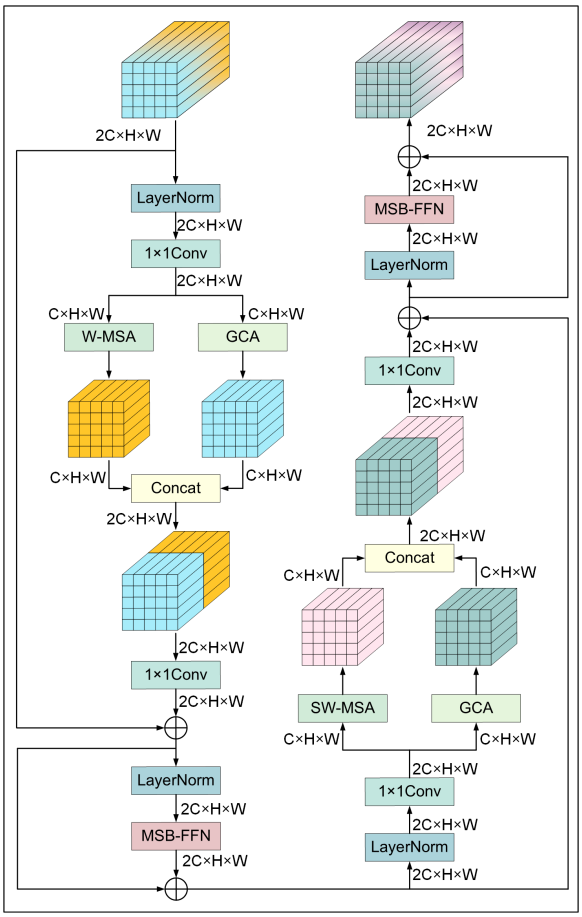

Fig. 5. Hyper-Prior Hybrid Attention Model. C denotes the number of channels, $C = 192$ when $\lambda = \{128, 256, 512\}$ and $C = 256$ when $\lambda = \{1024, 2048, 4096\}$. LayerNorm denotes layer normalization. 1×1Conv means that the convolution kernel size is 1, the step size is 1, and the number of channels is set to C. W-MSA denotes Window Multihead Self-Attention. GCA denotes Gated Channel Attention. MSB-FFN denotes a Multi-Scale and Multi-Branch Feed-Forward Network. SW-MSA denotes Shift Window Multihead Self-Attention.

3. Experiments

3.1 Datasets

For training and evaluating LTCHM, we employ the DOTA high-resolution remote sensing image dataset, the UC-Merced dataset, and the China Gaofen satellite dataset. The DOTA dataset, randomly selected from DOTA-v1 and DOTA-v2, consists of 5,489 high-resolution images with dimensions ranging from approximately 800×800 to 4000×4000 pixels. The UC-Merced dataset includes 2,100 remote sensing images covering 21 scene categories, each sized at 256×256 pixels. The China Gaofen satellite dataset comprises images from Gaofen-1 and Gaofen-5 satellites, totaling 2,531 images, each 256×256 pixels in size. In experiments, we randomly divide the three datasets into training, testing, and validation sets in an $8 : 1 : 1$ ratio.

3.2 Training Details

In the training process, we use the Adam optimizer [37] with a batch size of 8. Images are resized to 256×256 pixels through random cropping. The training experiences 200k iterations, beginning with an initial learning rate of 1×10^{-4}, which decreases to 1×10^{-5} after 100k iterations until the end of training. Using λ to control the rate-distortion trade-off of the model. The λ values are set to [128, 256, 512, 1024, 2048, 4096] to obtain different rate-distortion curves. The first three lambda values are for low-rate models, and the latter three are for high-rate models. The number of channels is set to 192 for low-rate models and 256 for high-rate models. Our compression model is implemented using the PyTorch framework and trained on an NVIDIA RTX 3090 GPU.

3.3 Traditional Codecs

For JPEG2000 implementation, we obtain the official OpenJPEG library from https://www.openjpeg.org/ and use the default configuration. Compression and decompression are performed using the following command:

```
opj_compress -i [input_file]
-o [output_file.j2k] -r [comression_ratio]
opj_decompress -i [input_file.j2k]
-o [output_file]
```

Where the compression_ratio is set to {20, 30, 40, 60, 110, 160}.

For the BPG implementation, we download the BPG software from http://bellard.org/bpg/ and use the default configuration. Compression and decompression are performed using the following command:

```
bpgenc -q [quality] [input_file]
-o [output.bpg]
bpgdec -o [output_file] [input.jpg]
```

Where the quality is set to {25, 30, 35, 40, 43, 46}.

We implemented AVIF and WebP using the third-party online tool Squoosh (https://squoosh.app/), with compression rates for AVIF set to {10, 20, 30, 40, 50, 60} and for WebP set to {1, 4, 20, 35, 60, 80}.

3.4 Architecture Details

Table 1 shows the detailed parameter settings for each component in the main encoder and decoder, and Table 2 lists the parameter configurations for each component in the hyper-prior encoder and decoder. More detailed parameter configuration can be found in our source code.

3.5 Evaluation Strategies

To measure the rate-distortion performance of the designed compression model, we employ PSNR, MS-SSIM, Learned Perceptual Image Patch Similarity (LPIPS), and Visual Information Fidelity in the pixel domain (VIFp) as metrics of measure distortion, with the bitrate expressed in bits per pixel (bpp). PSNR is an objective standard of image

	Input size	Layer	Output size
Encoder	$3 \times 256 \times 256$	[Conv|$K3$|C|$S2$ Conv|$K3$|C|$S1$, Conv|$K1$|C|$S2$ GDN]	$C \times 128 \times 128$
	$C \times 128 \times 128$	[DAM : C, K, G Conv|$K3$|C|$S1$ Conv|$K3$|C|$S1$]	$C \times 128 \times 128$
	$C \times 128 \times 128$	[Conv|$K3$|C|$S2$ Conv|$K3$|C|$S1$, Conv|$K1$|C|$S2$ GDN]	$C \times 64 \times 64$
	$C \times 64 \times 64$	[DAM : C, K, G Conv|$K3$|C|$S1$ Conv|$K3$|C|$S1$]	$C \times 64 \times 64$
	$C \times 64 \times 64$	[Conv|$K3$|C|$S2$ Conv|$K3$|C|$S1$, Conv|$K1$|C|$S2$ GDN]	$C \times 32 \times 32$
	$C \times 32 \times 32$	[DAM : C, K, G Conv|$K3$|C|$S1$ Conv|$K3$|C|$S1$]	$C \times 32 \times 32$
	$C \times 32 \times 32$	[Conv|$K3$|$2C$|$S2$ DAM : $2C, K, G$]	$2C \times 16 \times 16$
Decoder	$2C \times 16 \times 16$	[DAM : $2C, K, G$ Conv|$K3$|$2C$|$S1$ Conv|$K3$|C|$S1$]	$C \times 16 \times 16$
	$C \times 16 \times 16$	[Conv|$K3$|C|$S2$ Conv|$K3$|C|$S1$, Conv|$K1$|C|$S2$ IGDN]	$C \times 32 \times 32$
	$C \times 32 \times 32$	[DAM : C, K, G Conv|$K3$|C|$S1$ Conv|$K3$|C|$S1$]	$C \times 32 \times 32$
	$C \times 32 \times 32$	[Conv|$K3$|C|$S2$ Conv|$K3$|C|$S1$, Conv|$K1$|C|$S2$ IGDN]	$C \times 64 \times 64$
	$C \times 64 \times 64$	[DAM : C, K, G Conv|$K3$|C|$S1$ Conv|$K3$|C|$S1$]	$C \times 64 \times 64$
	$C \times 64 \times 64$	[Conv|$K3$|C|$S2$ Conv|$K3$|C|$S1$, Conv|$K1$|C|$S2$ IGDN]	$C \times 128 \times 128$
	$C \times 128 \times 128$	[Conv|$K3$|C|$S1$ Conv|$K3$|C|$S1$ Conv|$K3$|3|$S2$]	$3 \times 256 \times 256$

Tab. 1. The table lists the detailed parameters for each primary encoder and decoder component. $K3$ indicates a convolution kernel size of 3, while $K1$ represents a kernel size of 1. C denotes the number of channels, $C = 192$ when $\lambda = \{128, 256, 512\}$ and $C = 256$ when $\lambda = \{1024, 2048, 4096\}$. $S2$ denotes a stride of 2, and $S1$ indicates a stride of 1. DAM refers to the dynamic attention model. K specifies the convolution kernel size, which is set to $\{1, 3\}$ in the experiments, and G represents the number of groups, set to 2 in our experiments.

	Input size	Layer	Output size
Hyper-Encoder	$2C \times 16 \times 16$	[HPHAM : $2C, h32, win4$ Conv|$K3$|C|$S1$ Conv|$K3$|C|$S1$ Conv|$K3$|$2C$|$S2$]	$2C \times 8 \times 8$
	$2C \times 8 \times 8$	[HPHAM : $2C, h32, win4$ Conv|$K3$|C|$S1$ Conv|$K3$|C|$S2$]	$C \times 4 \times 4$
Hyper-Decoder	$C \times 4 \times 4$	[Conv|$K3$|C|$S2$ Conv|$K3$|$2C$|$S1$ HPHAM : $2C, h32, win4$]	$2C \times 8 \times 8$
	$2C \times 8 \times 8$	[Conv|$K3$|$1.5C$|$S1$ Conv|$K3$|$1.5C$|$S2$ Conv|$K3$|$2C$|$S1$ HPHAM : $2C, h32, win4$]	$2C \times 16 \times 16$

Tab. 2. The table shows the detailed parameters for each component in the hyper-prior encoder and decoder, where $h32$ refers to a head dimension of 32 and $win4$ refers to a window size of 4. HPHAM denotes Hyper-Prior Hybrid Attention Model.

quality that assesses the peak signal ratio between the original and reconstructed images, with higher values indicating better quality. MS-SSIM measures image similarity by combining structural and luminance information and considering variations at different scales. The MS-SSIM value ranges from –1 to 1, with values closer to 1 indicating higher similarity between the original and reconstructed images. MS-SSIM is particularly effective for natural images, aligning with human visual perception of images. When using MS-SSIM, it is necessary to convert the original MS-SSIM values to decibels (dB) using $-10\log_{10}(1-\text{MS-SSIM})$ for a more precise comparison. Bpp represents the average number of bits required to encode or compress each pixel of an image. LPIPS is a metric for evaluating perceptual similarity between images, commonly used in image generation, compression, and enhancement tasks. Unlike traditional metrics such as PSNR and SSIM, LPIPS leverages feature representations from deep learning networks to assess perceptual differences, aligning more with how the human visual system perceives images. LPIPS values range from 0 to 1, with lower values indicating higher perceptual similarity between images. A score below 0.1 usually signifies that the images are visually similar, while scores above 0.4 suggest noticeable differences. VIFp is rooted in information theory and suggests that the perceptual quality of an image can be quantified by the amount of information the human visual system extracts. VIFp measures image quality by comparing the visual information retained in both the original and distorted images. Its values typically range from 0 to 1, where 1 indicates perfect visual consistency between the compressed and original images, while values closer to 0 represent poorer image quality.

3.6 Rate-Distortion Performance

To validate the effectiveness of the proposed method, we compare LTCHM with four well-established traditional image compression standards, JPEG2000, BPG, AVIF and WebP, as well as with recent deep learning-based image compression models. These models include the discrete Gaussian mixture likelihood entropy model and simple attention model [26] (denoted as Cheng2020), the global reference entropy model [18] (denoted as Qian2020), the transformer-based entropy model [20] (denoted as Qian2022), the hybrid transformer-CNN image compression model [33] (denoted as Liu2023), and the region-adaptive transform-based image compression model [38] (denoted as Liu2024). We obtain the experimental data for the comparative methods by retraining and testing the provided open-source code on the DOTA, UC-Merced, and China Gaofen satellite datasets.

We evaluate the effects of LTCHM by comparing its rate-distortion performance on the DOTA dataset. Figure 6 displays the rate-distortion curves using PSNR, MS-SSIM, LPIPS and VIFp as image quality metrics. As depicted in Fig. 6, the proposed model achieves superior rate-distortion performance in both PSNR, MS-SSIM, LPIPS and VIFp compared to the methods of Cheng2020, Qian2020, Qian2022, Liu2023, and Liu2024. It also significantly surpasses traditional compression methods such as JPEG2000, BPG, AVIF and WebP in rate-distortion performance. Figures 7 and 8 show the rate-distortion curves for the UC-Merced and Gaofen satellite datasets, respectively. To comprehensively assess the compression performance of the proposed method, we conduct cross-dataset testing. The model is trained on the DOTA training set and tested on the UC-

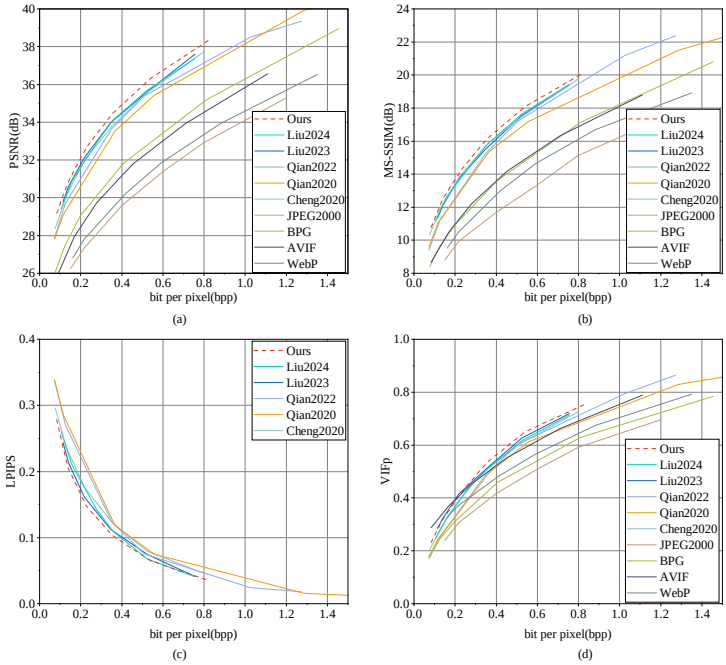

Fig. 6. Rate-distortion performance on the DOTA dataset. (a) Distortion measured by PSNR. (b) Distortion measured by MS-SSIM. (c) Distortion measured by LPIPS. (d) Distortion measured by VIFp.

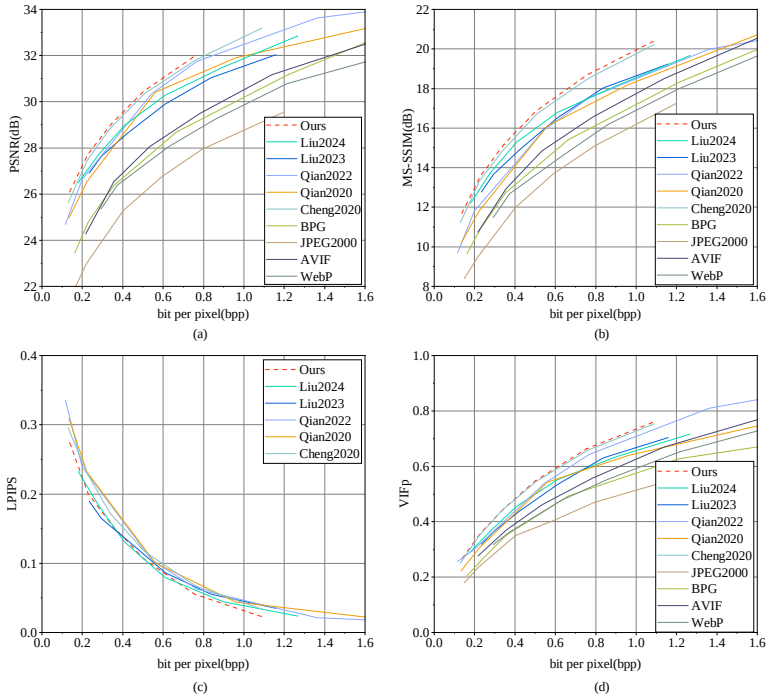

Fig. 7. Rate distortion performance on the UC-Merced dataset. (a) Distortion measured by PSNR. (b) Distortion measured by MS-SSIM. (c) Distortion measured by LPIPS. (d) Distortion measured by VIFp.

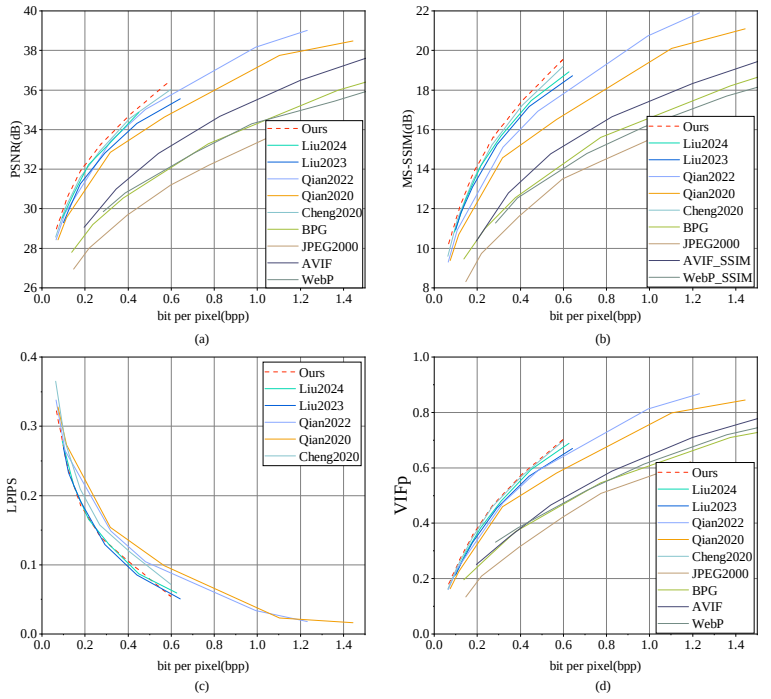

Fig. 8. Rate distortion performance on the China Gaofen satellite dataset. (a) Distortion measured by PSNR. (b) Distortion measured by MS-SSIM. (c) Distortion measured by LPIPS. (d) Distortion measured by VIFp.

Merced and a Gaofen satellite dataset. The rate-distortion curves of the test results are shown in Figs. 9 and 10. Furthermore, We also tested the model on a subset of 54 DOTA images, which were not part of the training data. This subset includes object types like buildings, vegetation, and urban and rural areas. The results are displayed in Fig. 11. LTCHM exceeds other comparative methods in PSNR and MS-SSIM, further validating its robustness and superior performance across different datasets.

In order to acquire quantitative results, we employ PSNR-BPP curves to calculate BD-rate [39] and BD-PSNR

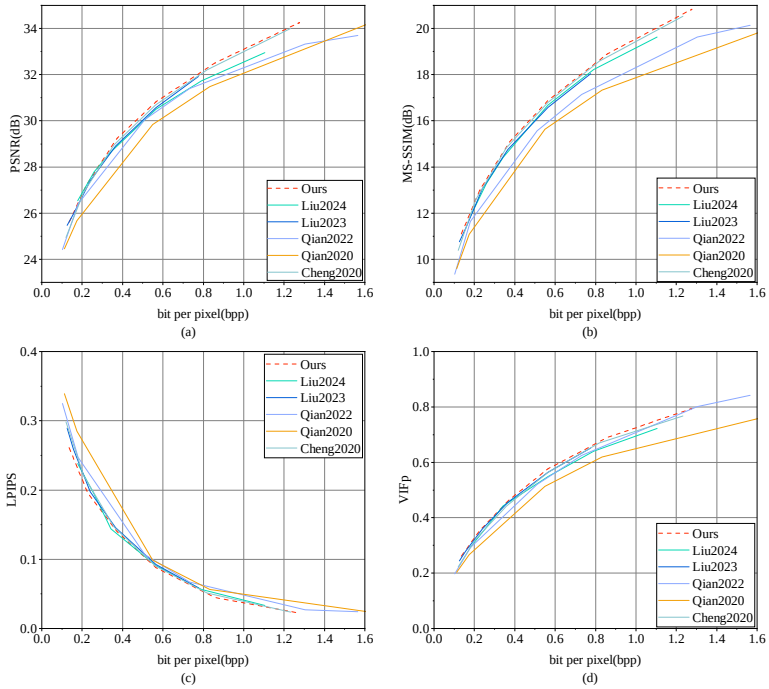

Fig. 9. Rate distortion performance is trained with the DOTA dataset and tested with the UC-Merced dataset. (a) Distortion measured by PSNR. (b) Distortion measured by MS-SSIM. (c) Distortion measured by LPIPS. (d) Distortion measured by VIFp.

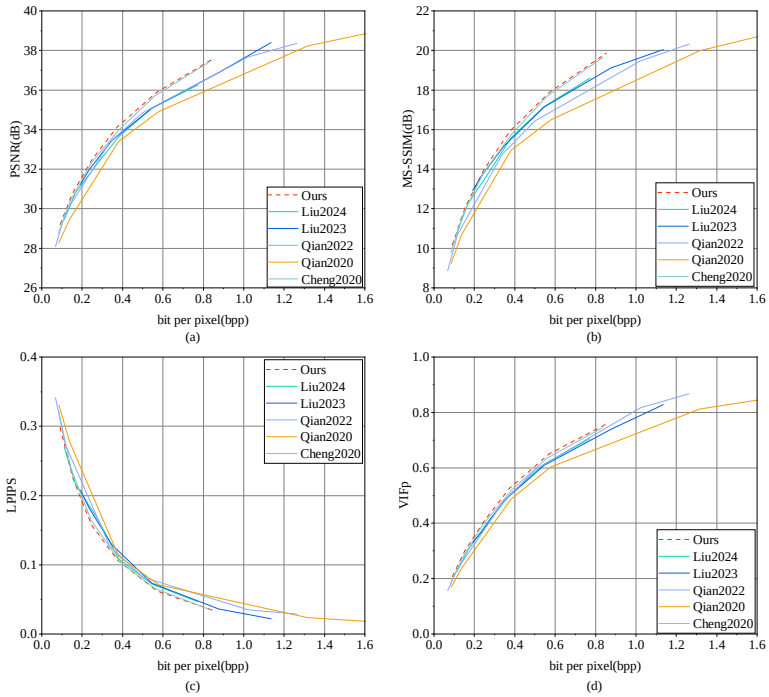

Fig. 10. Rate distortion performance is trained with the DOTA dataset and tested with the Gaofen satellite dataset. (a) Distortion measured by PSNR. (b) Distortion measured by MS-SSIM. (c) Distortion measured by LPIPS. (d) Distortion measured by VIFp.

as quantitative metrics. Using JPEG2000 as the anchor (BD-rate is equal to 0%) on the DOTA, UC-Merced, and China Gaofen satellite datasets, we compare the BD-rate and BD-PSNR results of LTCHM with other methods, as shown in Tab. 3. The data shows that LTCHM achieves bit rate savings of 72.693%, 69.499%, and 75.217% on the DOTA, UC-Merced, and China Gaofen satellite datasets, respectively, compared to JPEG2000, while increasing BD-PSNR by 5.131 dB, 4.350 dB, and 4.822 dB. Compared to the best-performing deep learning method, LTCHM shows improvements in BD-rate by 1.579%, 1.735%, and 1.948%, and in BD-PSNR by 0.289 dB, 0.144 dB, and 0.271 dB.

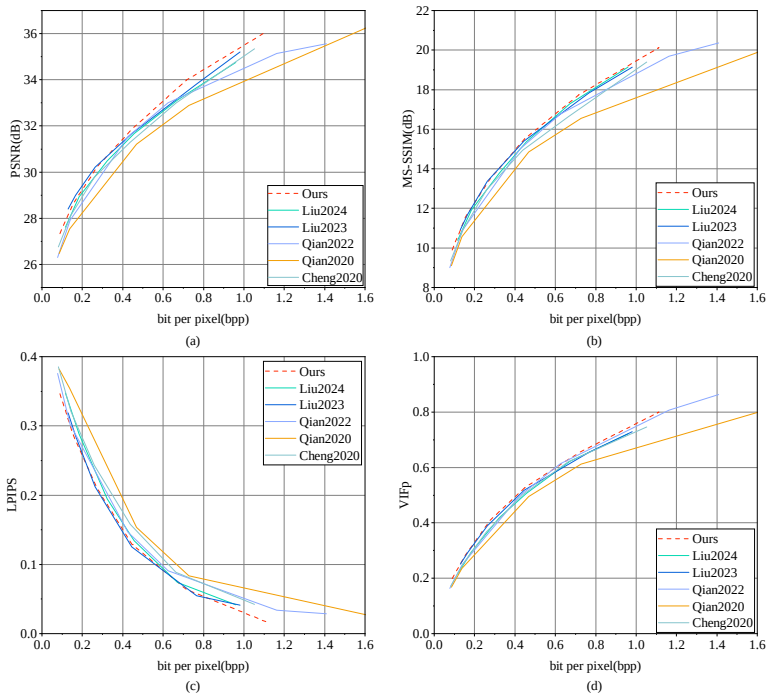

Fig. 11. The DOTA dataset is used for training and assesses the rate-distortion performance on a subset of images not included in the training. (a) Distortion measured by PSNR. (b) Distortion measured by MS-SSIM. (c) Distortion measured by LPIPS. (d) Distortion measured by VIFp.

Model	DOTA		UC-Merced		Gaofen satellite	
	BD-rate	BD-PSNR	BD-rate	BD-PSNR	BD-rate	BD-PSNR
BPG	–43.266%	2.054 dB	–33.674%	1.531 dB	–28.913%	1.010 dB
AVIF	–32.001%	1.626 dB	–34.635%	1.638 dB	–39.697%	1.850 dB
WebP	–11.117%	0.467 dB	–28.749%	1.214 dB	–27.495%	1.106 dB
Liu2024	–69.857%	4.764 dB	–66.889%	3.762 dB	–73.109%	4.282 dB
Liu2023	–71.114%	4.842 dB	–62.141%	3.324 dB	–71.418%	3.973 dB
Qian2022	–69.468%	4.471 dB	–65.358%	3.938 dB	–72.759%	4.304 dB
Qian2020	–66.744%	4.161 dB	–61.082%	3.711 dB	–69.669%	3.699 dB
Cheng2020	–70.504%	4.677 dB	–67.764%	4.206 dB	–73.269%	4.551 dB
Ours	**–72.693%**	**5.131 dB**	**–69.499%**	**4.350 dB**	**–75.217%**	**4.822 dB**

Tab. 3. The comparison of BD-rate and BD-PSNR results between the method proposed in this paper and other comparative methods on DOTA, UC-Merced, and China Gaofen satellite datasets highlights the best result in each column.

3.7 Visualization

We select one image from each of the DOTA, UC-Merced, and Gaofen satellite datasets, compress these images using different methods, and visualize the decompressed images to verify the visual superiority of LTCHM. Figures 12, 13, and 14 show the original images and the visual results of the comparison methods. In contrast, LTCHM preserves more details. This makes the reconstructed images clearer in texture details and object edges.

3.8 Ablation Study

To verify the effectiveness of DAM and HPHAM, we conduct ablation experiments on the DOTA dataset using the same parameters. The compression framework of Cheng2020 serves as the baseline, denoted as a baseline, without the simple attention module. DAM is added to the baseline, denoted as a baseline+DAM, to determine the impact of DAM on remote sensing image compression. We also incorporate HPHAM with and without GCA into the baseline to test whether HPHAM and channel information enhance rate-distortion performance, denoted as a baseline+HPHAM (w/ GCA) and baseline+HPHAM (w/o GCA), respectively. Combining DAM and HPHAM with GCA in the baseline, denoted as a baseline+DAM+HPHAM (w/ GCA), assesses their joint impact on rate-distortion performance. The rate-distortion curves in Fig. 15 demonstrate that baseline+DAM, baseline+HPHAM (w/ GCA), baseline+HPHAM (w/o GCA), and baseline+DAM+HPHAM (w/ GCA) all significantly outperform the baseline. Results confirm that DAM and HPHAM improve rate-distortion performance. Baseline+HPHAM (w/ GCA) outperforms base-

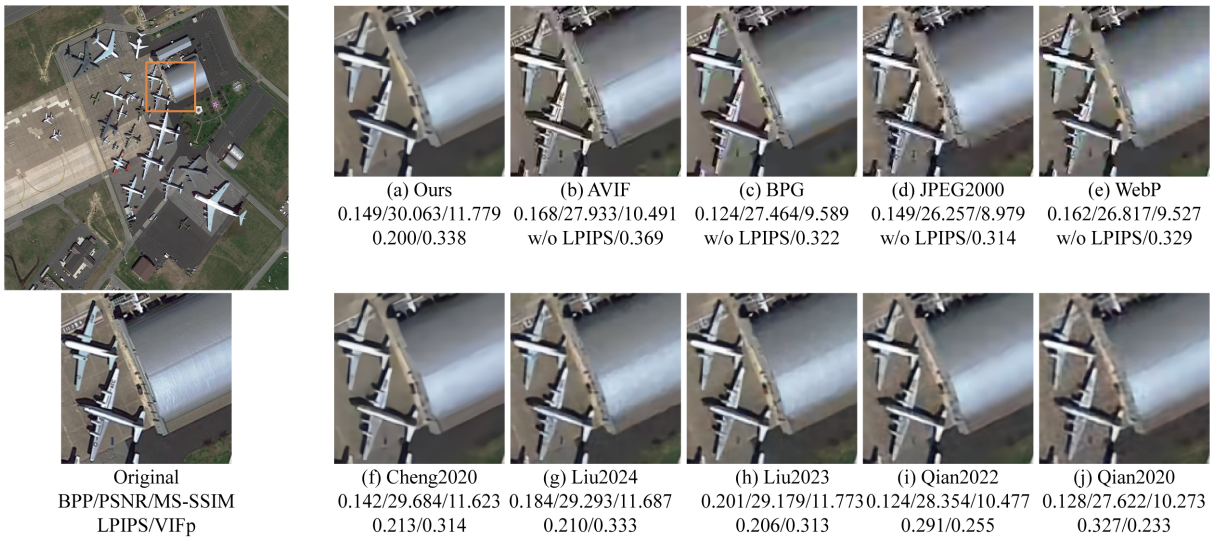

Fig. 12. The visualization results of a DOTA dataset image decompress with different image compression methods.

Fig. 13. The visualization results of a UC-Merced dataset image decompress with different image compression methods.

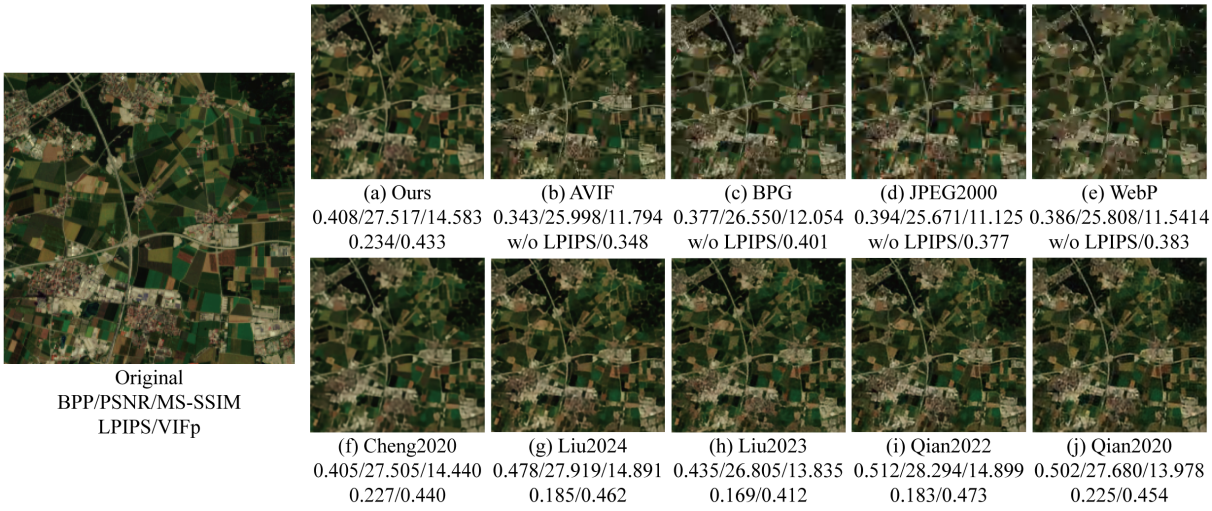

Fig. 14. The visualization results of a Gaofen satellite dataset image decompress with different image compression methods.

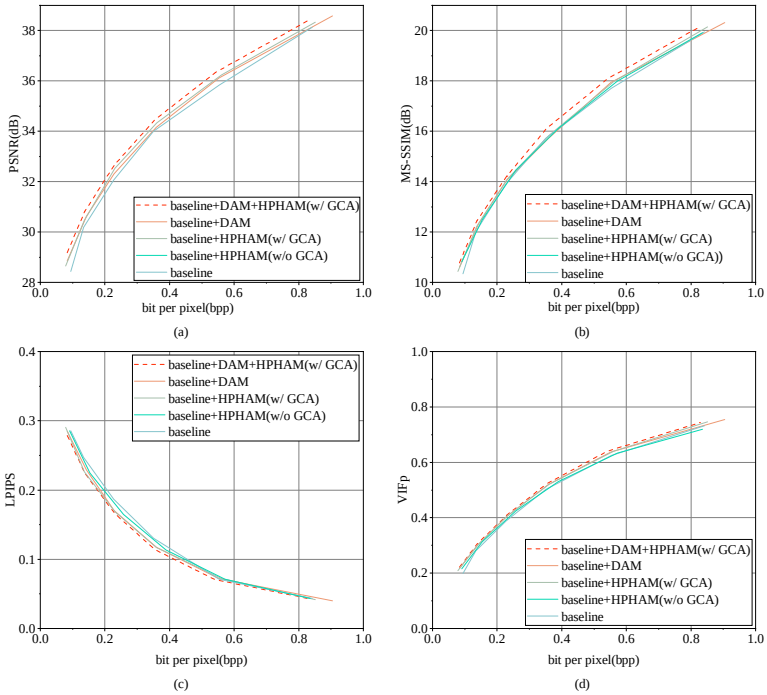

Fig. 15. Rate distortion performance based on ablation experiments with the DOTA dataset.

Model	Params (M)	BD-rate	BD-PSNR
baseline	7.43	0	0
baseling+DAM	8.41	–4.233%	0.187 dB
baseline+HPHAM(w/ GCA)	11.36	–5.621%	0.245 dB
baseline+HPHAM(w/o GCA)	11.24	–2.082%	0.114 dB
baseline+DAM+HPHAM(w/ GCA)	12.34	–9.658%	0.423 dB

Tab. 4. The number of parameters, BD-rate and BD-PSNR for different models.

line+HPHAM (w/o GCA) due to the ability of GCA to adjust attention weights based on channel information dynamically. This permits the model to focus accurately on critical features. HPHAM (w/ GCA) fuses channel and non-local information, which enhances rate-distortion performance. The result confirms the effectiveness of GCA and channel information in improving rate distortion. DAM adjusts attention dynamically based on local and global features. This enables the model to focus more effectively on critical local details and better capture global information. HPHAM (w/ GCA) leverages spatial and channel information of latent representations, which boosts image reconstruction and compression by considering spatial and inter-channel relationships. Therefore, baseline+DAM+HPHAM (w/ GCA) effectively combines local, non-local, and channel information for optimal rate-distortion performance.

Table 4 contains the number of parameters, BD-rate, and BD-PSNR for each model. DAM, HPHAM (w/ GCA), and HPHAM (w/o GCA) add 0.98M, 3.93M, and 3.81M parameters, respectively, representing 7.95%, 31.84%, and 30.87% of the total parameters. On the DOTA dataset, baseline+DAM, baseline+HPHAM (w/ GCA), and baseline+HPHAM (w/o GCA) reduce bitrates by 4.233%, 5.621%, and 2.082%, respectively. Combining DAM and HPHAM (w/ GCA), baseline+DAM+HPHAM (w/ GCA) performs best. The model can save 9.658% of bit rates and improve the reconstructed image quality by 0.423 dB.

3.9 Comparison of Various Attention Models

On the DOTA dataset, with the compression framework of Cheng2020 [26] serving as the baseline, we compare DAM with the Swin-Transformer-base Attention Model (SWAtten) [33], Simplified Attention Module (SAM) [26], and Window Attention Model (WAM) [17]. As illustrated in Fig. 16, DAM achieves the best rate-distortion performance because DAM can dynamically adjust attention to various regions, effectively focusing on key areas and critical features in the image.

Floating-point operations (FLOPs) and the number of parameters are employed to assess the complexity of different attention models. Table 5 lists the FLOPs and the number of parameters for DAM and other attention models, all with an input dimension of $8 \times 192 \times 128 \times 128$. The DAM has

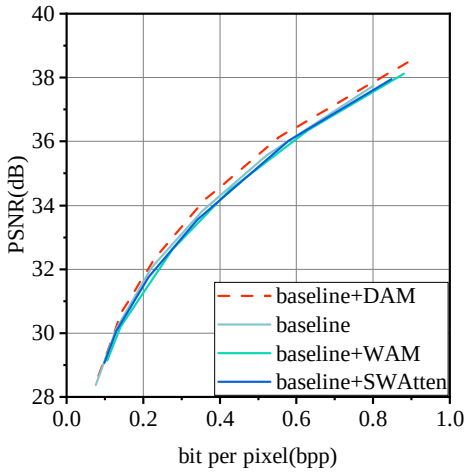

Fig. 16. Rate-distortion performance of different attention models.

Attention model	FLOPs (G)	Params (M)
SWAtten	102.27	0.78
SAM	99.05	0.76
WAM	19.53	0.15
DAM	**5.30**	**0.54**

Tab. 5. FLOPs and the number of parameters for different attention methods.

a significant advantage in FLOPs with only 5.30G, compared to the SWAtten, SAM, and WAM. However, the DAM has an increase of 0.39M in the number of parameters compared to WAM.

3.10 Complexity Analysis

The test set in the DOTA dataset is employed to evaluate the FLOPs, number of parameters, encoding time, and decoding time of the model. The input image is 256×256 pixels in size, and the test results are presented in Tab. 6. Since the hardware environment and input data influence encoding and decoding times, the times in Tab. 6 are averaged over all input. According to Tab. 6, LTCHM increases by 8.19G in FLOPs compared to the method of Qian2022 but remains significantly lower than Liu2023 and Liu2024.The entire compression framework of Liu2023 combines Swin-Transformer and CNN, which increase substantial computational overhead and complexity. The Segmentation-Prior-Guided Image Compression of Liu2024 includes two main modules: Region Adaptive Transformation (RAT) and Scale Affine Layer (SAL). RAT utilizes adaptive convolution to apply different kernels to various regions based on segmentation masks. This region-specific processing increases computational complexity as it may require separate calculations for each area. The RAT module utilizes depth-wise separable convolution, merging depth-wise and point-wise convolutions. Depth-wise separable convolution combines depth-wise and point-wise convolution as a new convolution in the RAT module.

Method	FLOPs (G)	Params (M)	Times (s)	
			Encoding	Decoding
Cheng2020	27.41	8.78	0.078	0.080
Qian2020	32.10	25.52	0.049	0.061
Qian2022	16.72	39.00	0.022	0.039
Liu2023	116.78	75.90	0.113	0.123
Liu2024	190.55	91.17	0.071	0.081
Ours	**24.91**	**12.34**	**0.124**	**0.127**

Tab. 6. FLOPs, number of parameters, encoding time and decoding time for different compression methods.

Although the computational complexity is similar to standard depth-wise separable convolution, its design may necessitate extra parameters and computations to generate and apply distinct kernels. Regarding the number of parameters, LTCHM increases by 3.56M compared to Cheng2020 but is lower than the other methods. By integrating the Swin-Transformer structure solely in the hyper-prior encoding and decoding and using DAM and residual blocks in the main encoder-decoder, LTCHM significantly reduces FLOPs and the number of parameters. Although LTCHM achieves lower FLOPs and parameter complexity, it has the longest encoding and decoding times relative to other methods. This is due to the complex matrix operations and weight computations required by DAM and HPHAM. Additionally, attention mechanisms, despite their theoretical parallelism, may suffer from low practical parallelization efficiency.

4. Conclusion

This paper proposes a new image compression framework called the Low-complexity Transformer-CNN Hybrid Model (LTCHM). The framework comprises two essential modules: the Dynamic Attention Model (DAM) and the Hyper-Prior Hybrid Attention Model (HPHAM). DAM boosts rate-distortion performance by adaptively adjusting attention weights to focus on crucial regions of the image. Experiments reveal that DAM surpasses previous attention models designed for image compression. Furthermore, HPHAM improves compression efficiency by incorporating Gated Channel Attention (GCA) into the swin-transformer. This integration enables parallel operation with W-MSA and SW-MSA. It effectively captures non-local and channel information in latent representations. GCA adjusts channel weights dynamically, strengthening HPHAM's ability to recognize and process different features and improve image reconstruction quality in complex scenes. Results show that our approach provides notable advantages in rate-distortion performance and visual quality on DOTA, UC-Merced, and China Gaofen satellite datasets, outperforming current image compression methods. As remote sensing technology advances, the diversity of collected data types has significantly increased, encompassing optical imagery, synthetic aperture radar (SAR) imagery, and light detection and ranging (LiDAR) data. Although existing compression methods perform well, they still need to improve generalization. In the future, we will explore how to effectively integrate multi-

modal data for compression and develop more adaptive deep learning models. These models automatically adjust compression rates based on different scenarios and resolutions, improving information utilization and compression efficiency to meet diverse application needs.

References

[1] WALLACE, G. K. The JPEG still picture compression standard. *Communications of the ACM*, 1991, vol. 34, no. 4, p. 30–44. DOI: 10.1145/103085.103089

[2] RABBANI, M., JOSHI, R. An overview of the JPEG 2000 still image compression standard. *Signal Processing: Image Communication*, 2002, vol. 17, no. 1, p. 3–48. DOI: 10.1016/S0923-5965(01)00024-8

[3] SULLIVAN, G. J., OHM, J. R., HAN, W. J., et al. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, vol. 22, no. 12, p. 1649–1668. DOI: 10.1109/TCSVT.2012.2221191

[4] GARCIA-VILCHEZ, F., SERRA-SAGRISTA, J. Extending the CCSDS recommendation for image data compression for remote sensing scenarios. *IEEE Transactions on Geoscience and Remote Sensing*, 2009, vol. 47, no. 10, p. 3431–3445. DOI: 10.1109/TGRS.2009.2021067

[5] TANG, Z., WANG, H., YI, X., et al. Joint graph attention and asymmetric convolutional neural network for deep image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022, vol. 33, no. 1, p. 421–433. DOI: 10.1109/TCSVT.2022.3199472

[6] AFJAL, M. I., UDDIN, P., MAMUN, A., et al. An efficient lossless compression technique for remote sensing images using segmentation based band reordering heuristics. *International Journal of Remote Sensing*, 2021, vol. 42, no. 2, p. 756–781. DOI: 10.1080/01431161.2020.1812130

[7] HONG, G., HALL, G., TERRELL, T. Discrete cosine transform data compression applied to satellite sensor images. *Remote Sensing*, 1995, vol. 16, no. 5, p. 835–850. DOI: 10.1080/01431169508954447

[8] ZHANG, L., QIU, B. Fast orientation prediction-based discrete wavelet transform for remote sensing image compression. *Remote Sensing Letters*, 2013, vol. 4, no. 12, p. 1156–1165. DOI: 10.1080/2150704X.2013.858838

[9] XIANG, X., JIANG, Y., SHI, B. Hyper-spectral image compression based on band selection and slant Haar type orthogonal transform. *International Journal of Remote Sensing*, 2024, vol. 45, no. 5, p. 1658–1677. DOI: 10.1080/01431161.2024.2318775

[10] SHI, C., ZHANG, J., ZHANG, Y. A novel vision-based adaptive scanning for the compression of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2015, vol. 54, no. 3, p. 1336–1348. DOI: 10.1109/TGRS.2015.2478145

[11] BALLE, J., LAPARRA, V., SIMONCELLI, E. P. End-to-end optimized image compression. *arXiv preprint*, 2016, p. 1–14. DOI: 10.48550/arXiv.1611.01074

[12] BALLE, J., MINNEN, D., SINGH, S., et al. Variational image compression with a scale hyperprior. *arXiv preprint*, 2018, p. 1–23. DOI: 10.48550/arXiv.1802.01436

[13] MINNEN, D., BALLE, J., TODERICI, G. D. Joint autoregressive and hierarchical priors for learned image compression. *Advances in Neural Information Processing Systems*, 2018, vol. 31, p. 1–22. DOI: 10.48550/arXiv.1809.02736

[14] WANG, Z., SIMONCELLI, E. P., BOVIK, A. C. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*. Pacific Grove (CA, USA), 2003, p. 1398–1402. DOI: 10.1109/ACSSC.2003.1292216

[15] HE, D., YANG, Z., PENG, W., et al. 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*. New Orleans (LA, USA), 2022, p. 5718–5727. DOI: 10.48550/arXiv.2203.10886

[16] XIE, Y., CHENG, K. L., CHEN, Q. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM International Conference on Multimedia*. Virtual Event China, 2021, p. 162–170. DOI: 10.1145/3474085.3475213

[17] ZOU, R., SONG, C., ZHANG, Z. 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*. New Orleans (LA, USA), 2022, p. 17492–17501. DOI: 10.48550/arXiv.2203.08450

[18] QIAN, Y., TAN, Z., SUN, X., et al. Learning accurate entropy model with global reference for image compression. *arXiv preprint*, 2020, p. 1–13. DOI: 10.48550/arXiv.2010.08321

[19] CHEN, T., LIU, H., MA, Z., et al. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 2021, vol. 30, p. 3179–3191. DOI: 10.1109/TIP.2021.3058615

[20] QIAN, Y., LIN, M., SUN, X., et al. Entroformer: A transformer-based entropy model for learned image compression. *arXiv preprint*, 2022, p. 1–14. DOI: 10.48550/arXiv.2202.05492

[21] KOYUNCU, A. B., GAO, H., BOEV, A., et al. Contextformer: A transformer with spatio-channel attention for context modeling in learned image compression. In *European Conference on Computer Vision*. Tel Aviv (Israel), 2022, p. 447–463. DOI: 10.1007/978303119800726

[22] FU, C., DU, B. Remote sensing image compression based on the multiple prior information. *Remote Sensing*, 2023, vol. 15, no. 8, p. 1–16. DOI: 10.3390/rs15082211

[23] ZHANG, L., PAN, T., HUANG, Y., et al. SAR image compression using discretized Gaussian adaptive model and generalized subtractive normalization. *IEEE Geoscience and Remote Sensing Letters*, 2022, vol. 19, p. 1–5. DOI: 10.1109/LGRS.2022.3213375

[24] ZHANG, J., ZHANG, S., WANG, H., et al. Image compression network structure based on multiscale region of interest attention network. *Remote Sensing*, 2023, vol. 15, no. 2, p. 1–18. DOI: 10.3390/rs15020522

[25] ZHANG, L., HU, X., PAN, T., et al. Image compression network structure based on multiscale region of interest attention network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023, vol. 17, p. 138–149. DOI: 10.1109/JSTARS.2023.3326957

[26] CHENG, Z., SUN, H., TAKEUCHI, M., et al. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle (WA, USA), 2020, p. 7939–7948. DOI: 10.1109/CVPR42600.2020.00796

[27] LIN, F., SUN, H., LIU, J., et al. Multistage spatial context models for learned image compression. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Rhodes Island (Greece), 2023, p. 1–5. DOI: 10.1109/ICASSP49357.2023.10095875

[28] FU, H., LIANG, F., LIN, J., et al. Learned image compression with gaussian-laplacian-logistic mixture model and concatenated residual modules. *IEEE Transactions on Image Processing*, 2023, vol. 32, p. 2063–2076. DOI: 10.1109/TIP.2023.3263099

[29] KINGMA, D. P., WELLING, M. Auto-encoding variational bayes. *arXiv preprint*, 2013, p. 1–14. DOI: 10.48550/arXiv.1312.6114

[30] DOSOVITSKIY, A., BEYER, L., KOLESNIKOV, A., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint*, 2020, p. 1–22. DOI: 10.48550/arXiv.2010.11929

[31] LIU, Z., LIN, Y., CAO, Y., et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Montreal (QC, Canada), 2021, p. 10012–10022. DOI: 10.1109/ICCV48922.2021.00986

[32] KHOSHKHAHTINAT, A., ZAFARI, A., MEHTA, P., et al. Multi-context dual hyper-prior neural image compression. In *International Conference on Machine Learning and Applications (ICMLA)*. Jacksonville (FL, USA), 2023, p. 618–625. DOI: 10.1109/ICMLA58977.2023.00091

[33] LIU, J., SUN, H., KATTO, J. Learned image compression with mixed transformer-CNN architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Vancouver (BC, Canada), 2023, p. 14388–14397. DOI: 10.1109/CVPR52729.2023.01383

[34] LOU, M., ZHOU, H. Y., YANG, S., et al. TransXNet: Learning both global and local dynamics with a dual dynamic token mixer for visual recognition. *arXiv preprint*, 2023, p. 1–12. DOI: 10.48550/arXiv.2310.19380

[35] NARAYANAN, M. SENetV2: Aggregated dense layer for channelwise and global representations. *arXiv preprint*, 2023, p. 1–9. DOI: 10.48550/arXiv.2311.10807

[36] GAO, S. H., CHENG, M. M., ZHAO, K., et al. Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, vol. 43, no. 2, p. 652–662. DOI: 10.1109/TPAMI.2019.2938758

[37] KINGMA, D. P., BA, J. Adam: A method for stochastic optimization. *arXiv preprint*, 2014, p. 1–15. DOI: 10.48550/arXiv.1412.6980

[38] LIU, Y., YANG, W., BAI, H., et al. Region-adaptive transform with segmentation prior for image compression. *arXiv preprint*, 2024, p. 1–19. DOI: 10.48550/arXiv.2403.00628

[39] BJONTEGAARD, G. Calculation of average PSNR differences between RD-curves. *ITU SG16 Doc. VCEG-M33*, 2001.

About the Authors . . .

Lili ZHANG received her B.E., M.E., and Ph.D. degrees from Jilin University, Changchun, China, in 2002, 2005, and 2012. She is currently an Associate Professor at the College of Electronic Information Engineering, Shenyang Aerospace University, Shenyang, China. Her current research interests include image compression, radar-based human activity classification, and SAR-based ship detection.

Xianjun WANG received a B.E. degree from Shenyang Aerospace University, Shenyang, China, in 2020. He is currently a postgraduate student at the College of Electronic Information Engineering, Shenyang Aerospace University, Shenyang, China. His current research interests include deep learning and image compression.

Jiahui LIU received a B.E. degree from Shenyang Aerospace University, Shenyang, China, in 2023. He is currently a postgraduate student at the College of Electronic Information Engineering, Shenyang Aerospace University, Shenyang, China. His current research interests include deep learning and image compression.

Qizhi FANG received a B.E. degree from Shenyang Aerospace University, Shenyang, China, in 2002 and received a M.E. degree from Northeastern University, Shenyang, China, in 2008. He is currently an Associate Professor at the College of Electronic Information Engineering, Shenyang Aerospace University and Liaoning General Aviation Academy, Shenyang, China. His current research interests include image compression, radar-based human activity classification, and SAR-based ship detection.