Learned Image Compression with Dictionary-based Entropy Model

Jingbo Lu[1] Leheng Zhang[1] Xingyu Zhou[1] Mu Li[2] Wen Li[1] Shuhang Gu[1]
[1] University of Electronic Science and Technology of China
[2] Harbin Institute of Technology, Shenzhen
{jingbolu2023, shuhanggu}@gmail.com
https://github.com/LabShuHangGU/DCAE

Abstract

Learned image compression methods have attracted great research interest and exhibited superior rate-distortion performance to the best classical image compression standards of the present. The entropy model plays a key role in learned image compression, which estimates the probability distribution of the latent representation for further entropy coding. Most existing methods employed hyper-prior and auto-regressive architectures to form their entropy models. However, they only aimed to explore the internal dependencies of latent representation while neglecting the importance of extracting prior from training data. In this work, we propose a novel entropy model named Dictionary-based Cross Attention Entropy model, which introduces a learnable dictionary to summarize the typical structures occurring in the training dataset to enhance the entropy model. Extensive experimental results have demonstrated that the proposed model strikes a better balance between performance and latency, achieving state-of-the-art results on various benchmark datasets.

1. Introduction

Image compression is a vital and well-established research area in the field of image signal processing. The substantial demand for high-resolution images requires powerful compression techniques to address storage and transmission challenges. Classical standards such as JPEG [43], JPEG2000 [40], and VVC [38] have been widely adopted over time, following a general pipeline: transforming, quantization, and entropy coding. Recently, the learned image compression (LIC) methods [13, 15, 17, 20, 21, 25, 27, 32, 44, 51, 52] have demonstrated outstanding performance, even surpassing the current best image and video coding standards VVC.

Learned image compression primarily consists of two key components: nonlinear auto-encoder and entropy model. The encoder and decoder play the role of transforming the image and the latent representation into each other; while, the entropy model estimates the probability distribution of the latent representation for further entropy coding [30, 31]. In the last several years, for the pursuit of better compression performance, one category of studies [11, 25, 27, 46, 51, 52] investigates advanced auto-encoder structures to establish delicate latent representation extractors. While, another line of research [6, 11, 15, 16, 22, 24, 32–34, 36, 37] pays attention to the more fundamental and unique component of LIC, i.e. the entropy model, to optimize the rate and distortion (RD) trade-off.

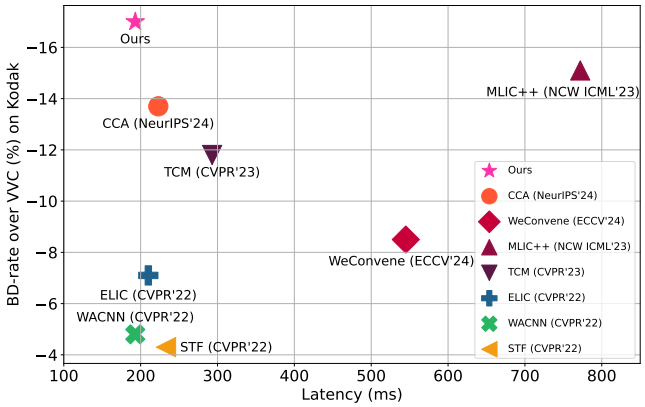

Figure 1. Rate-speed comparison on Kodak. Left-top is better.

At the core of the entropy model in LIC framework is a distribution estimator. In their seminal work, Ballé *et al.* [5] showed that the smallest code length of latent representation is given by the cross entropy between a parameterized distribution predictor (entropy model) and the real distribution of latents, and used a fully factorized density model to capture the probability distributions of the latent representation. Subsequently, Ballé *et al.* [6] and Minnen *et al.* [34] respectively introduced the hyper-prior and the auto-regressive frameworks, which leverage side-information or decoded representation to provide image dependent priors for better capturing the distribution of latent representation. The great success achieved by the hyperprior and the autoregressive framework has inspired numerous follow-up

works, introducing priors to establish a conditional entropy model has become a prevailing strategy in the literature of LIC. In the last several years, numerous attempts have been made in establishing elaborate causal context models [11, 16, 22, 24, 32, 33, 36, 37] as well as designing sophisticated network architectures [11, 25, 27, 46, 51, 52] for better exploiting internal dependency among latent representation.

In this paper, instead of further exploring advanced internal dependency modeling architectures, we propose a Dictionary-based Cross Attention Entropy model (DCAE) which tries to take additional advantages from external training data to improve the entropy model in the LIC framework. In the existing LIC works, training data helps the entropy model indirectly by providing examples to learn functions for capturing internal dependency, little attention has been paid to directly extracting priors from training data to boost the performance of entropy model. Nevertheless, the training data contains prior of natural images, which has proven to be effective in providing supplementary information to recover the image from corrupted observations in the field of image restoration. Therefore, exploiting external prior for boosting entropy modeling could be a promising direction. In order to leverage typical patterns in natural image, we learn a dictionary from the training dataset to summarize typical structures. Then, during the auto-regressive predicting process, decoded representation, which contain partial information of local structure, can work as query tokens to select similar dictionary entries to predict the remaining representation in a cross-attention manner. In addition, we utilize feature maps from different convolutional layers to achieve multi-scale texture extraction for helping to achieve more accurate dictionary queries. These innovations enable our model to outperform existing state-of-the-art methods. Our contributions can be summarized as follows:

- We apply a dictionary to summarize useful information from the training dataset. During the auto-regressive predicting process, we can use the decoded representation, which contains partial information of local structure, to select similar dictionary entries to assist in predicting the remaining representation in a cross-attention manner.
- We leverage features with different receptive fields to capture textures at various scales, allowing the model to capture fine-grained and coarse-level texture information, which helps us perform more precise dictionary queries.
- Experiments show that our method achieves state-of-the-art performance regarding coding performance and running speed(Fig.1 and Tab. 1).

2. Related Work

2.1. Learned Image Compression

Learned nonlinear transforms. The powerful nonlinear transformation capability is one of the keys to the learned image compression. It functions as an encoder or decoder, converting input images into compact latent representation for entropy coding, or transforming latent representation back into reconstructed images. The development of learned nonlinear transforms can be categorized into two types: CNN-based methods and transformer-based methods. Since Ballé *et al.* [7] first proposed the GDN to reduce the mutual information between features, the combination of GDN and CNN has been widely used in subsequent methods [6, 8, 34]. In order to further enhance the nonlinear transformation capability of the CNN, Cheng *et al.* [11] adopted the attention module to deal with the challenging content, whereas Xie *et al.* [46] proposed the enhanced invertible encoding network to mitigate the problem of information loss. Recently, numerous efforts have been made to explore the transformer architecture [42]. Zou *et al.* [52] and Zhu *et al.* [51] first employed the Swin transformer layer [29] to construct their encoder and decoder. Zou *et al.* [52] further proposed a window attention module to enhance their CNN-based model by focusing on spatially neighboring elements. Liu *et al.* [27] incorporated CNN and transformer as a fundamental module. Furthermore, Li *et al.* [25] utilized different window sizes of the Swin transformer layer to capture various frequency information. Inspired by [18] and [26], we use residual blocks and residual Swin transformer blocks to construct our encoder and decoder for capturing local and non-local information simultaneously. Additionally, we further improve our transformer by using advanced techniques such as ResScale [48] and Convolutional Gated Linear Unit [39].

Entropy Model. Another crucial module in learned image compression is the entropy model, which is used to evaluate the distribution parameters of latent representation for entropy coding. Existing entropy models primarily pay attention to the investigation of how to capture the internal dependencies among latent representation. Ballé *et al.* [6] first proposed the hyper-prior to capture spatial internal dependencies of the latent representation. Minnen *et al.* [34] then proposed the serial auto-regressive context model that utilizes the adjacent decoded latent representation to assist the distribution estimating of current latent representation. In order to capture long-range dependencies of latent representation, Qian *et al.* [36] used the most relevant latent representation and Kim *et al.* [22] utilized the attention mechanism of transformer to capture global information. Furthermore, to address the issue of slow encoding and decoding in serial auto-regressive context models, some studies [16, 17, 32, 33] set the internal dependencies of the auto-regressive context model to a fixed order to achieve parallelization. In addition, different sophisticated network architectures [24, 25, 27, 37] were also designed to enhance the ability to model internal dependencies. Despite various advancements aimed at improving the modeling capabilities of entropy model, they

mainly focused on utilizing the internal dependencies of latent representation and neglected the exploration of prior information in external training data. Different from these works, the proposed Dictionary-based Cross Attention Entropy model concentrates on capturing typical patterns and textures from training dataset to establish external dependencies between the latent representation and prior derived from training dataset. Therefore, our entropy model can achieve more accurate distribution estimation for entropy coding.

2.2. Dictionary Learning

Dictionary learning has demonstrated powerful potential in the fields of image generation and image restoration due to its ability of effectively utilizing prior information from the training dataset. In image generation tasks, Van *et al.* [41] first introduced a dictionary to generate clear images. In order to achieve visually pleasing generation quality, Esser *et al.* [12] employed perceptual and adversarial losses to train the dictionary. In image restoration tasks, Gu *et al.* [14] used a dictionary for face restoration. Liu *et al.* [28] further learned a set of basis dictionaries from different types of datasets for obtaining more flexible and expressive prior. In addition, Zhang *et al.* [50] employed a dictionary to study various cluster centers, enabling self-attention operations on tokens of the same category. In learned image compression, Minnen *et al.* [35] first utilized a non-learnable dictionary from K-means++ algorithm [3] to improve entropy model. In addition, Kim *et al.* [22] utilized eight learnable tokens that need to be transmitted to capture global internal dependencies. Since its tokens are derived from the image itself and a limited number of tokens need to be transmitted, this restricts its ability to improve the entropy model. In this paper, we propose a Dictionary-based Cross Attention Entropy model, where we employ learnable shared network parameters as the dictionary for encoding and decoding to capture external dependencies, allowing us to perform more accurate distribution estimation.

3. Method

3.1. Formulation

The overall pipeline of the learned image compression model is shown in Fig. 2. In the encoding stage, given an input image $\boldsymbol{x}$, the encoder g_a first transforms it into a latent representation $\boldsymbol{y}$: $\boldsymbol{y} = g_a(\boldsymbol{x})$. The entropy model is then used to estimate the distribution parameters $\boldsymbol{\Phi} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ of latent representation $\boldsymbol{y}$ for entropy coding. According to previous studies [17, 34], $\boldsymbol{y}$ is discretized to $\hat{\boldsymbol{y}} = \lceil \boldsymbol{y} - \boldsymbol{\mu} \rfloor + \boldsymbol{\mu}$ through quantization, and $\lceil \boldsymbol{y} - \boldsymbol{\mu} \rfloor$ is further losslessly encoded into bitstreams based on the evaluated distribution parameters $\boldsymbol{\Phi}$. In the decoding stage, after restoring $\hat{\boldsymbol{y}}$ from the bitstreams, the decoder g_s reconstructs the high-quality image $\hat{\boldsymbol{x}}$ from the quantized latent representation $\hat{\boldsymbol{y}}$: $\hat{\boldsymbol{x}} = g_s(\hat{\boldsymbol{y}})$.

In this process, the entropy model plays one of the most important roles, which determines the encoding length of the latent representation $\boldsymbol{y}$. Most existing entropy models employ the hyper-prior architecture [6] and the channel-wise auto-regressive architecture [33]. For the hyper-prior architecture, a side information $\boldsymbol{z} = h_a(\boldsymbol{y})$ is first introduced to capture the internal spatial dependencies in the latent representation $\boldsymbol{y}$. Then, the hyper-prior decoder h_s maps $\hat{\boldsymbol{z}}$ to latent feature $\boldsymbol{\mathcal{F}}_z$ for estimating the distribution $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ of latent representation $\boldsymbol{y}$: $\boldsymbol{\mathcal{F}}_z = h_s(\hat{\boldsymbol{z}})$. Conditioned on $\hat{\boldsymbol{z}}$, the latent representation $\boldsymbol{y}$ is modeled as a joint Gaussian distribution: $p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) = \left[\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2})\right](\hat{\boldsymbol{y}})$. For channel-wise auto-regressive architecture, $\boldsymbol{y}$ is first divided into several even slices $\{\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_{s-1}\}$ along the channel dimension. These slices are then encoded and decoded in sequential order. The decoded slices $\overline{\boldsymbol{y}}_{<i} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_{i-1}\}$ are used to supplement information for encoding or decoding the subsequent slice $\boldsymbol{y}_i$, due to their similarity relationships. After obtaining the hyper-prior feature $\boldsymbol{\mathcal{F}}_z$ and the decoded slices $\overline{\boldsymbol{y}}_{<i}$, they are entered into the entropy module f_E to estimate $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ of $\hat{\boldsymbol{y}}_i$ for further encoding and decoding. Moreover, in order to compensate for the information loss caused by quantization, $\boldsymbol{\mathcal{F}}_z$, $\overline{\boldsymbol{y}}_{<i}$, and $\hat{\boldsymbol{y}}_i$ are leveraged to predict the quantization error $\boldsymbol{r}_i = \boldsymbol{y}_i - \hat{\boldsymbol{y}}_i$ through the latent residual prediction net f_{LRP} [33]. This process can be formulated as:

$$\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i = f_E(\boldsymbol{\mathcal{F}}_z, \overline{\boldsymbol{y}}_{<i}), \quad \boldsymbol{r}_i = f_{LRP}(\boldsymbol{\mathcal{F}}_z, \overline{\boldsymbol{y}}_{<i}, \hat{\boldsymbol{y}}_i). \quad (1)$$

To train the learned image compression model, a Lagrangian multiplier-based rate-distortion optimization is employed as the loss function:

$$\mathcal{L} = \mathcal{R}(\hat{\boldsymbol{y}}) + \mathcal{R}(\hat{\boldsymbol{z}}) + \lambda \cdot \mathcal{D}(\boldsymbol{x}, \hat{\boldsymbol{x}}), \quad (2)$$

where $\mathcal{R}(\hat{\boldsymbol{y}})$ and $\mathcal{R}(\hat{\boldsymbol{z}})$ denote the bitrates of $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$; $\mathcal{D}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ denotes the distortion between the input image $\boldsymbol{x}$ and reconstructed image $\hat{\boldsymbol{x}}$; λ controls the trade-off between rate and distortion.

3.2. Dictionary-based Cross Attention Entropy Model

Existing methods adopt hyper-prior and auto-regression frameworks to estimate the probability distribution of the latent representation $\boldsymbol{y}$. Both types of methods essentially utilize the internal dependencies within latent representation to model probability distribution, but neither explicitly leverages common patterns and textures in natural images as prior information for entropy estimation.

Our goal is to share a dictionary that preserves typical textures between the encoder and decoder. When estimating the distribution of latent representation, we can leverage the finite information to model the latent representation more accurately by querying the dictionary that retains complete

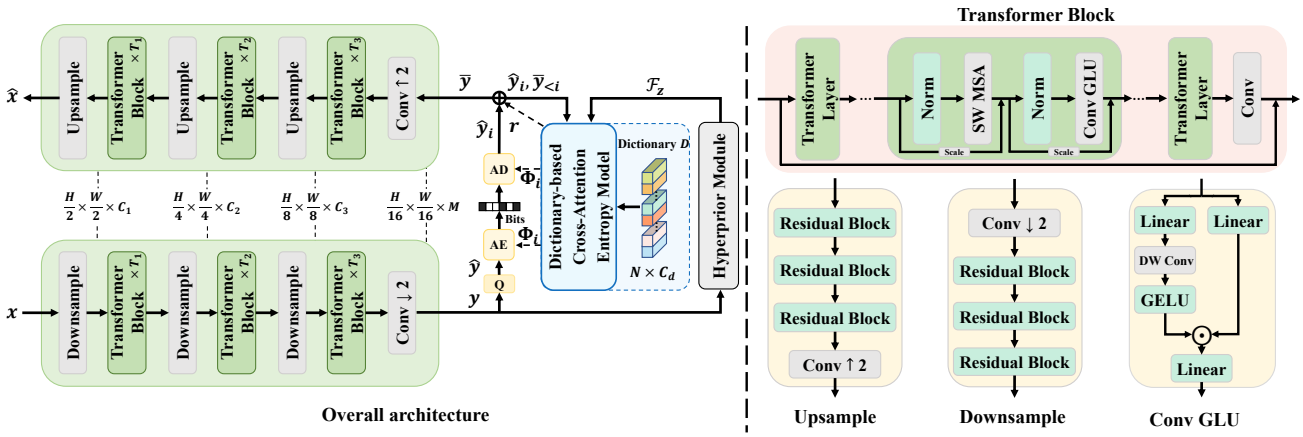

Figure 2. The overall framework of the proposed network. Given an input image $\boldsymbol{x}$, the encoder g_a transforms it into the latent representation $\boldsymbol{y}$, then the proposed dictionary-based cross-attention entropy model is used to encode or decode the quantized $\hat{\boldsymbol{y}}$. Finally, the decoder g_s reconstructs the image $\hat{\boldsymbol{x}}$ from the latent representation $\overline{\boldsymbol{y}}$. In the dictionary-based cross-attention entropy model, we introduce a learnable dictionary to capture typical structures and textures in natural images for improving the distribution estimation of the latent representation $\boldsymbol{y}$.

information. To achieve this goal, we propose a **D**ictionary-based **C**ross **A**ttention **E**ntropy model (**DCAE**) as shown in Fig. 3. Specifically, we first propose the Multi-Scale Features Aggregation module to obtain the multi-scale features $\boldsymbol{X}_{ms_i}$, which leverages features with different receptive fields to help capture textures at various scales, enabling more precise dictionary queries. We then propose the Dictionary-based Cross Attention module to extract the dictionary features $\boldsymbol{\mathcal{F}}_{dict_i}$ that utilizes learnable network parameters to constitute our dictionary $\boldsymbol{D}$, employing cross attention to dynamically query the complete information stored in the dictionary using the available partial texture information:

$$\begin{aligned} \boldsymbol{X}_{ms_i} &= \mathrm{MSFA}(\boldsymbol{X}_i), \\ \boldsymbol{\mathcal{F}}_{dict_i} &= \mathrm{DCA}(\boldsymbol{X}_{ms_i}, \boldsymbol{D}), \end{aligned} \tag{3}$$

where $X_i = [\boldsymbol{\mathcal{F}}_z, \overline{\boldsymbol{y}}_{<i}]$. Finally, different from Eq. 1, which does not explicitly use prior information in natural images, we combine the dictionary feature $\boldsymbol{\mathcal{F}}_{dict_i}$ with hyper-prior feature $\boldsymbol{\mathcal{F}}_z$ and channel-wise auto-regressive feature $\overline{\boldsymbol{y}}_{<i}$ and put them into the entropy module f_E to obtain the parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ of the Gaussian distribution. Furthermore, $\boldsymbol{\mathcal{F}}_{dict_i}$, $\boldsymbol{\mathcal{F}}_z$, $\overline{\boldsymbol{y}}_{<i}$ and $\hat{\boldsymbol{y}}_i$ are input to the latent residual prediction net f_{LRP} [33] to predict the quantization error $\boldsymbol{r}_i$:

$$\begin{aligned} \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i &= f_E(\boldsymbol{\mathcal{F}}_z, \overline{\boldsymbol{y}}_{<i}, \boldsymbol{\mathcal{F}}_{dict_i}), \\ \boldsymbol{r}_i &= f_{LRP}(\boldsymbol{\mathcal{F}}_z, \overline{\boldsymbol{y}}_{<i}, \boldsymbol{\mathcal{F}}_{dict_i}, \hat{\boldsymbol{y}}_i). \end{aligned} \tag{4}$$

Multi-Scale Features Aggregation Module. The proposed Multi-Scale Features Aggregation Module is used to capture multi-scale textures from feature maps, enabling more accurate querying of prior information stored in the dictionary. In Multi-Scale Features Aggregation Module, we first utilize feature maps from different convolutional layers to achieve multi-scale texture extraction. Feature maps from shallow convolutional layers have limited receptive fields, enabling them to capture finer textures. As the convolution depth increases, deeper layers can capture larger-scale textures. In addition, to achieve efficient convolutional computation, each basic convolution unit consists of two linear layers and a 3×3 depthwise (DW) convolution:

$$\begin{aligned} \mathrm{EConv}(\boldsymbol{X}) &= \mathrm{DWConv}_{3\times3}(\boldsymbol{X}\boldsymbol{W}^{in})\boldsymbol{W}^{out}, \\ \boldsymbol{X}_i^1 = \boldsymbol{X}_i, \quad \boldsymbol{X}_i^j &= \mathrm{EConv}(\boldsymbol{X}_i^{j-1}), \\ \boldsymbol{X}_i^{merge} &= [\boldsymbol{X}_i^1, ..., \boldsymbol{X}_i^m]\boldsymbol{W}^{merge}, \end{aligned} \tag{5}$$

where $\boldsymbol{W}^{in}, \boldsymbol{W}^{out}$ are used to perform the transformation of convolution channels and $\boldsymbol{W}^{merge}$ is used to merge multi-scale information; m represents the number of stacked efficient convolutional layers. To enable more precise queries of prior information from the dictionary, we then employ spatial attention module [45] to dynamically assign a weight to each spatial position.

$$\mathrm{MSFA}(\boldsymbol{X}_i) = \mathrm{SA}(\boldsymbol{X}_i^{merge}) \odot X_i^{merge}. \tag{6}$$

where $\mathrm{SA}(\boldsymbol{X}_i^{merge}) \in \mathbb{R}^{H\times W}$ represents the spatial weight map output by the spatial attention module.

Dictionary-based Cross Attention Module. After introducing how we utilize features with different receptive fields to capture textures at various scales, we will introduce how to use the existing information to perform dictionary queries.

We first established a shared dictionary $\boldsymbol{D}$ using learnable network parameters to preserve the common textures found in natural images. The dictionary $\boldsymbol{D}$ is initialized as a tensor with the shape of $[N, C_d]$, where N is the number of dictionary entries and C_d is the number of feature dimensions. During the training process, the dictionary will gradually

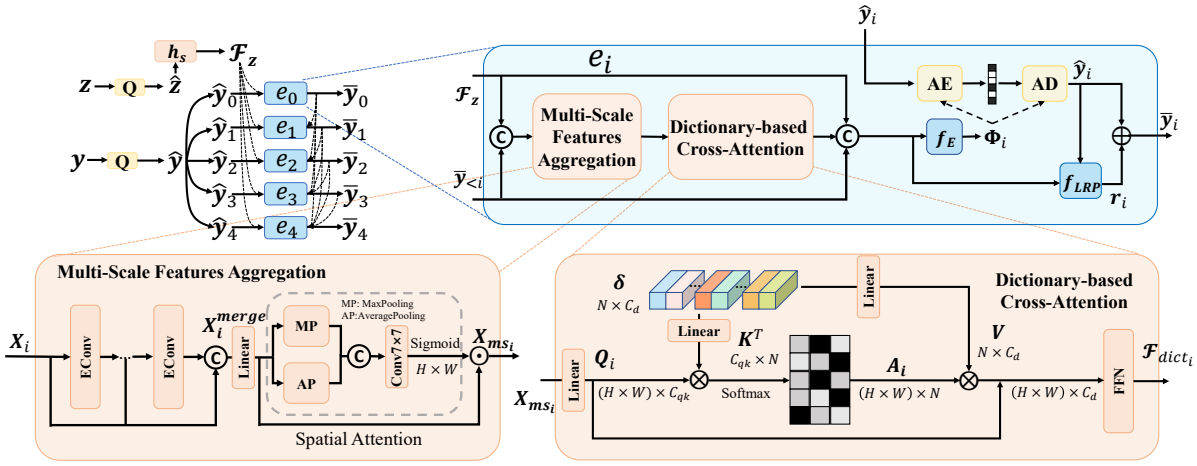

Figure 3. The proposed Dictionary-based Cross Attention Entropy model. The Dictionary-based Slice Network e_i is used to encode or decode the latent representation $\hat{\boldsymbol{y}}_i$. In e_i, the hyper-prior feature $\boldsymbol{\mathcal{F}}_z$ and channel-wise auto-regressive feature $\overline{\boldsymbol{y}}_{<i}$ are first fed into our Multi-Scale Features Aggregation module to obtain multi-scale features $\boldsymbol{X}_{ms_i}$. Then, the multi-scale features $\boldsymbol{X}_{ms_i}$ are used to query the dictionary to extract the dictionary feature $\boldsymbol{\mathcal{F}}_{dict_i}$. Finally, the dictionary feature $\boldsymbol{\mathcal{F}}_{dict_i}$ is taken as input to the entropy module f_E to estimate the distribution parameters Φ_i of $\hat{\boldsymbol{y}}_i$ for entropy coding, and to the latent residual prediction net f_{LRP} to predict the quantization error $\boldsymbol{r}_i$.

learn to fit the typical structures, which is similar to traditional dictionary learning methods [47, 49] learning to fit natural images with the image patch dictionary. In addition, it is noteworthy that this dictionary is simultaneously shared between the encoder and decoder, thereby not requiring additional bitrate for transmission.

We then apply cross attention to query the learnable dictionary with features that contain partial texture information, aiming to capture prior information present in natural images. Specifically, we use the multi-scale feature $\boldsymbol{X}_{ms_i}$ to generate the query tokens $\boldsymbol{Q}_i$, which contains partial texture information. The learnable dictionary $\boldsymbol{D}$ is used to generate the key tokens $\boldsymbol{K}$ and the value tokens $\boldsymbol{V}$. The key tokens $\boldsymbol{K}$ are utilized for calculating similarity with the query tokens, while the value tokens $\boldsymbol{V}$ represent the texture information stored in the dictionary.:

$$\boldsymbol{Q}_i = \boldsymbol{X}_{ms_i}\boldsymbol{W}^Q,\ \boldsymbol{K} = \boldsymbol{D}\boldsymbol{W}^K,\ \boldsymbol{V} = \boldsymbol{D}, \tag{7}$$

where $\boldsymbol{W}^Q \in \mathbb{R}^{C_{ms}\times C_{qk}}$, $\boldsymbol{W}^K \in \mathbb{R}^{C_d\times C_{qk}}$ are linear transforms for $\boldsymbol{X}_{ms}$ and $\boldsymbol{D}$, respectively. Then, we can obtain the dictionary feature $\boldsymbol{\mathcal{F}}_{dict}$ in a cross-attention manner:

$$\boldsymbol{A}_i = \mathrm{SoftMax}(\boldsymbol{Q}_i\boldsymbol{K}^T/\boldsymbol{\tau}), \quad \boldsymbol{\mathcal{F}}_{dict_i} = \boldsymbol{A}_i\boldsymbol{V}, \tag{8}$$

where $\boldsymbol{\tau}_i$ is a learnable scaling parameter to adjust the range of the dot product of $\boldsymbol{Q}_i$ and $\boldsymbol{K}$. $\boldsymbol{Q}_i\boldsymbol{K}^T \in \mathbb{R}^{(H\times W)\times N}$ represents the similarity map between query feature and dictionary entries. The softmax function is applied along the dimension N to normalize the weights of each dictionary entry. Finally, we utilize the normalized weights $\boldsymbol{A}_i$ to perform a weighted aggregation $\boldsymbol{A}_i\boldsymbol{V}$ of the dictionary information, resulting in the dictionary feature $\boldsymbol{\mathcal{F}}_{dict_i}$ and the $\boldsymbol{\mathcal{F}}_{dict_i}$ is further enhanced by FFN layer.

3.3. Network Architecture

Our main encoder g_a and decoder g_s is designed as Fig. 2 shows. We apply basic downsampling/upsampling modules followed by Swin transformer blocks [26] to perform non-linear mapping for extracting compact latent representation. Each basic downsampling / upsampling module consists of a strided / transposed convolution, along with several cascaded residual blocks to extract the local context information. The Swin transformer blocks are used to capture long-range dependencies, supplying non-local information. In addition, in order to further enhance non-linear transformation capability of our transformer, inspired by [48] and [39], we employ ResScale [48] to scale up transformer model size from depth and we adopt Convolutional Gated Linear Unit [39] to construct our FFN.

Previous methods [13, 17, 27], typically use the same number of channels and modules across different stages of the encoder-decoder. However, computing high-resolution features slows down the overall model speed. To achieve a more efficient structural design, we adopt varying numbers of channels and modules at different stages, shifting computations to lower-resolution stages to enable faster encoding and decoding speed. A more detailed description of this design can be found in Section 4.1.

4. Experiments

4.1. Experimental Settings

Training Details. We follow the experimental settings of recent state-of-the-art methods [52] and utilize the OpenImages dataset [19] to train our final model. OpenImages

Table 1. Computation burden and performance comparisons between different methods.

Model	Latency (ms)			GFLOPs		Params	BD-rate		
	Tot.	Enc.	Dec.	Enc.	Dec.		Kodak	Tecnick	CLIC
STF (CVPR'22) [52]	233	102	131	143	**161**	99.8M	-4.3%	-	-4.1%
WACNN (CVPR'22) [52]	193	**80**	113	138	231	75.0M	-4.8%	-	-4.4%
ELIC (CVPR'22) [17]	210	91	119	138	233	**41.9M**	-7.1%	-	-
M2T (ICCV'23) [32]	-	-	-	-	-	-	-8.5%	-	-
MT (ICCV'23) [32]	-	-	-	-	-	-	-12.5%	-	-
TCM (CVPR'23) [27]	293	142	151	307	441	75.9M	-11.8%	-12.0%	-12.0%
MLIC+ (ACMMM'23) [21]	-	-	-	-	-	-	-13.1%	-17.3%	-16.4%
MLIC++ (NCW ICML'23) [21]	772	362	410	222	300	116.5M	-15.1%	-18.6%	-16.9%
FTIC (ICLR'24) [25]	-	-	-	**127**	355	71.0M	-14.6%	-15.1%	-13.6%
WeConvene (ECCV'24) [13]	545	275	271	702	320	105.5M	-8.5%	-9.2%	-10.1%
CCA (NeurIPS'24) [15]	223	122	101	277	394	64.9M	-13.7%	-15.3%	-14.5%
Ours	**193**	93	**100**	252	305	119.2M	**-17.0%**	**-21.1%**	**-19.7%**
VVC	-	-	-	-	-	-	0%	0%	0%

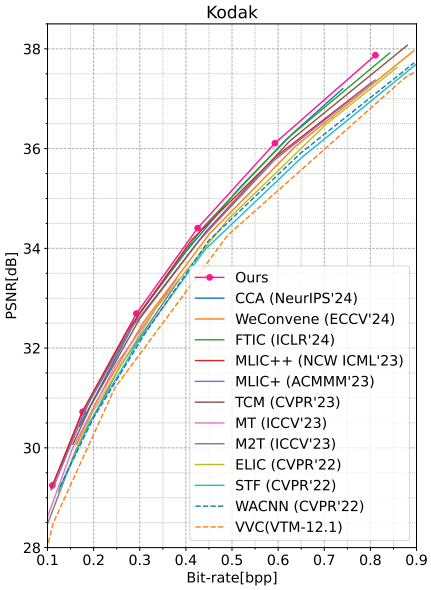

Figure 4. Performance evaluation (PSNR) on the Kodak dataset.

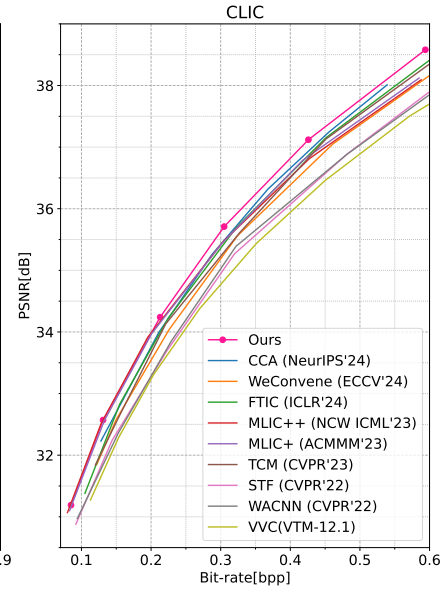

Figure 5. Performance evaluation (PSNR) on the CLIC dataset.

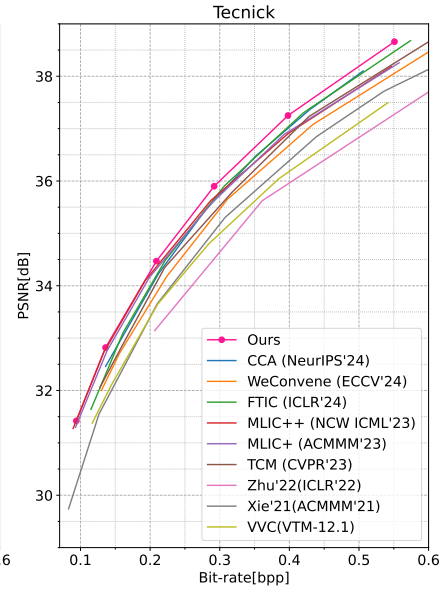

Figure 6. Performance evaluation (PSNR) on the Tecnick dataset.

dataset contains 300k images with short edge no less than 256 pixels. We randomly crop patches of size 256 × 256 for each training iteration, with a batch size of 16, and adopt the Adam optimizer [23] to minimize the R-D loss in Eq.2. We utilize Mean Squared Error (MSE) loss as the distortion measure to train our models. In order to obtain compression models with different compression ratio, we train our models with different λ values, *i.e.*, $\lambda = \{0.0018, 0.0035, 0.0067, 0.0130, 0.0250, 0.0500\}$. We train our models with a initial learning rate $1e-4$ for 80 epochs (1.5 million iterations), and then decrease the learning rate to $1e-5$ and train the models for another 20 epochs (0.375 million iterations) for obtaining the final models. We use RTX 4090 to complete our experiments.

Implementation Details. For our model, we introduce a learnable dictionary with 128 dictionary entries and 640 channels. We set the number of transformer layers within Transformer Block at different scales as $(T_1, T_2, T_3) = (1, 2, 12)$ for our model; where in the hyper-prior module, our model contain 1 transformer layer. As for the feature dimension, we set the feature dimension for different scales as $(C_1, C_2, C_3) = (96, 144, 256)$. The dimensions of the latent representation $\boldsymbol{y}$ and side information $\boldsymbol{z}$ are set to 320 and 192, respectively. The head dimensions of transformer layers in encoder g_a and decoder g_s are set as {8, 16, 32, 32, 16, 8}, while the head dimensions of transformer layers in hyper-prior encoder h_a, hyper-prior decoder h_s and Dictionary-based Cross Attention are set 32. The window

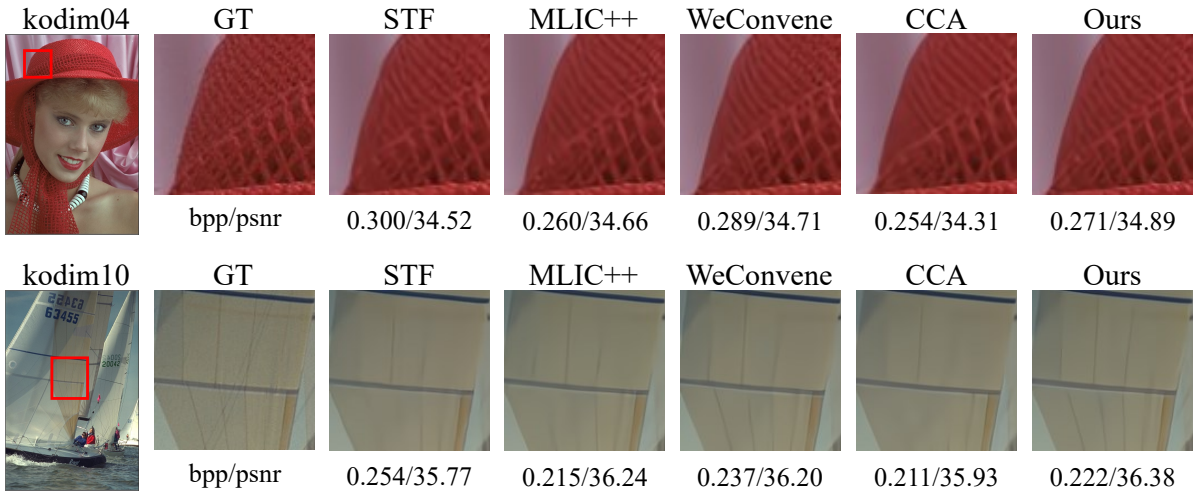

Figure 7. Reconstructed images *kodim04* and *kodim10* from the Kodak dataset. In the above visualization, our model effectively restores the texture information while maintaining comparable or lower bitrate.

sizes of these transformer layers are set as 8×8 for encoder g_a and decoder g_s and 4×4 for the hyper-prior module.

Comparison Methods and Benchmark Datasets. We choose three benchmark datasets, *i.e.*, Kodak image set [1], Tecnick testset [4], CLIC professional validation dataset [2] to evaluate our methods. The competing appraches including classical standard VVC (VTM-12.1) [38] and recent state-of-the-art LIC models [11, 13, 15, 17, 20, 21, 25, 27, 32, 46, 51, 52]. VVC results are achieved by CompressAI [9], while the results of other methods are provided by the method authors.

4.2. Ablation Studies

In this part, we conduct experiments to validate the effectiveness of the proposed Dictionary-based Cross-Attention Entropy (DCAE) model. We use a smaller model for ablation studies, where the number of transformer layers is set to $(T_1, T_2, T_3) = (0, 0, 4)$. All ablation studies are trained with a initial learning rate $1e-4$ for 20 epochs (0.75 million iterations) with a batch size of 8, followed by 5 epochs (0.1875 million iterations) of training with a learning rate of $1e-5$.

Table 2. Ablation studies of the proposed modules.

Model	BD-rate	Latency (ms)
baseline	-4.20%	143
+ DCA	-7.28%	153
+ MSFA	-8.50%	160

Effects of DCA and MSFA. In order to show the effectiveness of the proposed methods, we remove all proposed modules to establish a baseline and progressively add them back to demonstrate the benefits they provide. As shown in Tab.2, the DCA achieves a 3.08% improvement in BD-rate on kodak dataset while only increasing the coding latency of baseline model from 143 ms to 153 ms. Furthermore, MSFA further improves the BD rate from 7.28% to -8.50%.

Table 3. Ablation studies of the dictionary size.

N	-	64	128	192	256
BD-rate	-4.20%	-6.84%	**-7.28%**	-7.26%	-6.92%
Latency (ms)	143	153	153	154	154

Effects of dictionary size. In order to analyze the effect of dictionary size, we also train models with 64, 192 and 256 dictionary items, respectively. The BD-rate values on the Kodak dataset by different models are reported in Tab.3. When equipped with the DCA module, even with a smaller number of dictionary entries, *i.e.*, 64, the performance improves significantly, with the BD-rate improving from -4.20% to -6.84%. As the number of dictionary entries further increases to 128, the performance of the model improves further, reaching -7.28%. However, when the number of dictionary entries increases again, the improvement become saturated and does not result in better compression performance.

Table 4. Ablation studies of the MSFA.

m	0	1	2	3	4
BD-rate	-7.28%	-7.62%	-8.04%	**-8.50%**	-8.36%
Latency (ms)	153	157	158	160	162

Effects of the number of convolutional layers in MSFA. Increasing the number of convolutional layers m in MSFA will aid in modeling multi-scale features and enlarging the receptive field, thereby facilitating accurate dictionary queries.

Tab.4 shows the impact of the number of convolutional layers. Increasing the number of convolutional layers m from 1 to 3 improves the performance of our model, achieving a peak of -8.50% when m is 3. Further increasing m does not lead to any additional performance gains.

Comparison with Global Token. Kim *et al.* [22] proposed the global token to capture the global internal dependencies of latent representation. Both the global token and our dictionary use learnable network parameters to improve the entropy model. However, since the global token must generate distinct tokens for each image, it necessitates the transmission of these tokens during both encoding and decoding processes. In contrast, our dictionary captures common textures across different images, enabling its shared usage between the encoder and decoder. In addition, since the global token utilizes a relatively small number of tokens, whereas our dictionary employs a significantly larger number of dictionary entries (128 vs. 8), our dictionary can demonstrate superior representational capacity. To ensure a fair comparison, we apply the global token to our baseline, with the results presented in Tab.5. It can be observed that, under comparable latency (153 vs. 152), our DCA achieves better performance (-7.28% vs. -6.59%).

Table 5. Comparison with global token.

Model	BD-rate	Latency (ms)
baseline	-4.20%	143
DCA	-7.28%	153
global token	-6.59%	152

4.3. Comparisons with State-of-the-Art Methods

The rate-distortion performance by different methods on Kodak dataset, CLIC dataset, and Tecnick dataset are shown in Fig. 4, Fig. 5, and Fig. 6, which use PSNR to evaluate performance. Our proposed method consistently outperform the existing methods on all the three benchmark datasets. Additionally, we present the BD-rate results, GFLOPs and the compression latency information by our method and the current state-of-the-art methods in Tab. 1. The RD-rate [10] value is calculated with VVC (VTM-12.1) as the anchor. The latency and GFLOPs are calculated on the Kodak dataset. As can be found in the table, compared to MLIC++, which currently achieves the best BD-rate performance, our model outperforms it across all three datasets. Notably, the latency of MLIC++ on Kodak dataset is nearly four times that of our model. A more detailed Rate-speed comparison can be found in Fig. 1, which clearly demonstrates that our model is able to achieve good compression results with a smaller latency. Some visual examples by our proposed model as well as recent state-of-the-art methods are shown in Fig. 7. The visual results clearly validate the superiority of our model in keeping image details.

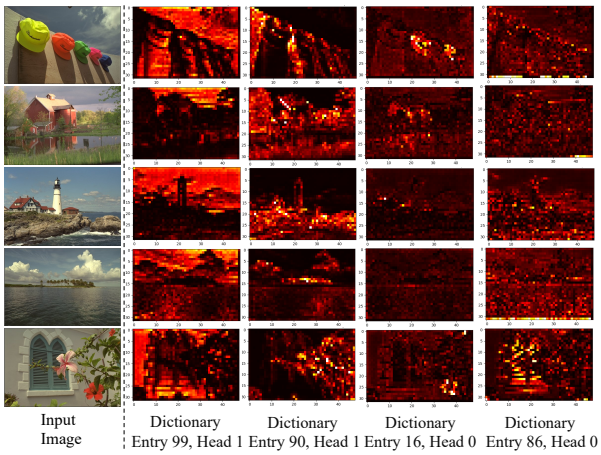

Figure 8. Visualization of attention maps between feature maps and dictionary entries. The first column represents original images from the Kodak dataset and the last four columns represent attention maps of a specific dictionary entry across different images.

4.4. Visualization Analysis

In order to analyze our argument of leveraging typical local structures, we present some intermediate attention maps to analyze the behaviour of our model. In Fig. 8, we present the same attention maps on different testing images. We can clearly observe corelations between dictionary items and image local structures, similar image local structures tend to leverage the same dictionary item to predict the latent representation. The results validate our idea of exploiting dictionary to provide prior information of typical structures.

5. Conclusion

In this paper, we propose a novel dictionary-based cross-attention entropy (DCAE) model for explicitly capturing prior information from the training dataset. The proposed entropy model uses learnable network parameters to summarize the typical structures and textures in natural images, thereby improving the entropy model. We show that DCAE brings effective improvement in RD performance. By incorporating the proposed DCAE, we exceed the state-of-the-art RD performance on three different resolution datasets (*i.e.*, Kodak, Tecnick, CLIC Professional Validation).

6. Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 62476051, No. 62176047) and Sichuan Natural Science Foundation (No. 2024NSFTD0041).

References

[1] Eastman kodak. kodak lossless true color image suite (photocd pcd0992). http://r0k.us/graphics/kodak/, 1993. 7

[2] Workshop and challenge on learned image compression and multi-class image classification. https://www.compression.cc/, 2020. 7

[3] David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, pages 1027–1035, 2007. 3

[4] Nicola Asuni and Andrea Giachetti. Testimages: a large-scale archive for testing visual devices and basic image processing algorithms. 2014. 7

[5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1

[6] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 1, 2, 3

[7] Johannes Ballé, Valero Laparra, and EeroP. Simoncelli. Density modeling of images using a generalized normalization transformation. *arXiv: Learning,arXiv: Learning*, 2015. 2

[8] Johannes Ballé, Valero Laparra, and EeroP. Simoncelli. End-to-end optimized image compression. *International Conference on Learning Representations,International Conference on Learning Representations*, 2016. 2

[9] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020. 7

[10] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. 2001. 8

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948, 2020. 1, 2, 7

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 3

[13] Haisheng Fu, Jie Liang, Zhenman Fang, Jingning Han, Feng Liang, and Guohe Zhang. Weconvene: Learned image compression with wavelet-domain convolution and entropy model. *arXiv preprint arXiv:2407.09983*, 2024. 1, 5, 6, 7

[14] Yuchao Gu, Xintao Wang, Liangbin Xie, Chao Dong, Gen Li, Ying Shan, and Ming-Ming Cheng. Vqfr: Blind face restoration with vector-quantized dictionary and parallel decoder. In *European Conference on Computer Vision*, pages 126–143. Springer, 2022. 3

[15] Minghao Han, Shiyin Jiang, Shengxi Li, Xin Deng, Mai Xu, Ce Zhu, and Shuhang Gu. Causal context adjustment loss for learned image compression. *arXiv preprint arXiv:2410.04847*, 2024. 1, 6, 7

[16] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021. 1, 2

[17] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5718–5727, 2022. 1, 2, 3, 5, 6, 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[19] Neil Alldrin Andreas Veit Sami Abu-El-Haija Serge Belongie David Cai Zheyun Feng Vittorio Ferrari Victor Gomes Abhinav Gupta Dhyanesh Narayanan Chen Sun Gal Chechik Ivan Krasin, Tom Duerig and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. https://github.com/openimages, 2016. 5

[20] Wei Jiang and Ronggang Wang. Mlic++: Linear complexity multi-reference entropy modeling for learned image compression. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023. 1, 7

[21] Wei Jiang, Jiayu Yang, Yongqi Zhai, Peirong Ning, Feng Gao, and Ronggang Wang. Mlic: Multi-reference entropy model for learned image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7618–7627, 2023. 1, 6, 7

[22] Jun-Hyuk Kim, Byeongho Heo, and Jong-Seok Lee. Joint global and local hierarchical priors for learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5992–6001, 2022. 1, 2, 3, 8

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[24] A Burakhan Koyuncu, Han Gao, Atanas Boev, Georgii Gaikov, Elena Alshina, and Eckehard Steinbach. Contextformer: A transformer with spatio-channel attention for context modeling in learned image compression. In *European Conference on Computer Vision*, pages 447–463. Springer, 2022. 1, 2

[25] Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Frequency-aware transformer for learned image compression. *arXiv preprint arXiv:2310.16387*, 2023. 1, 2, 6, 7

[26] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 2, 5

[27] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14388–14397, 2023. 1, 2, 5, 6, 7

[28] Kechun Liu, Yitong Jiang, Inchang Choi, and Jinwei Gu. Learning image-adaptive codebooks for class-agnostic image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5373–5383, 2023. 3

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[30] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, page 620–636, 2003. 1

[31] G Nigel N Martin. Range encoding: an algorithm for removing redundancy from a digitised message. In *Proc. Institution of Electronic and Radio Engineers International Conference on Video and Data Recording*, 1979. 1

[32] Fabian Mentzer, Eirikur Agustsson, and Michael Tschannen. M2t: Masking transformers twice for faster decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5340–5349, 2023. 1, 2, 6, 7

[33] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2020. 2, 3, 4

[34] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018. 1, 2, 3

[35] David Minnen, George Toderici, Saurabh Singh, Sung Jin Hwang, and Michele Covell. Image-dependent local entropy models for learned image compression. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 430–434. IEEE, 2018. 3

[36] Yichen Qian, Zhiyu Tan, Xiuyu Sun, Ming Lin, Dongyang Li, Zhenhong Sun, Hao Li, and Rong Jin. Learning accurate entropy model with global reference for image compression. *arXiv preprint arXiv:2010.08321*, 2020. 1, 2

[37] Yichen Qian, Ming Lin, Xiuyu Sun, Zhiyu Tan, and Rong Jin. Entroformer: A transformer-based entropy model for learned image compression. *arXiv preprint arXiv:2202.05492*, 2022. 1, 2

[38] K.R. Rao and Humberto Ochoa Dominguez. Versatile video coding, 2022. 1, 7

[39] Dai Shi. Transnext: Robust foveal visual perception for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17773–17783, 2024. 2, 5

[40] David S. Taubman. Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, page 286, 2002. 1

[41] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 3

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[43] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991. 1

[44] Dezhao Wang, Wenhan Yang, Yueyu Hu, and Jiaying Liu. Neural data-dependent transform for learned image compression. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 17379–17388, 2022. 1

[45] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 4

[46] Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM international conference on multimedia*, pages 162–170, 2021. 1, 2, 7

[47] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010. 5

[48] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2, 5

[49] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, pages 711–730. Springer, 2012. 5

[50] Leheng Zhang, Yawei Li, Xingyu Zhou, Xiaorui Zhao, and Shuhang Gu. Transcending the limit of local window: Advanced super-resolution transformer with adaptive token dictionary. *arXiv preprint arXiv:2401.08209*, 2024. 3

[51] Yinhao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *International Conference on Learning Representations*, 2021. 1, 2, 7

[52] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17492–17501, 2022. 1, 2, 5, 6, 7