Joint Global and Local Hierarchical Priors for Learned Image Compression

Jun-Hyuk Kim[1]* Byeongho Heo[2] Jong-Seok Lee[1]
[1]School of Integrated Technology, Yonsei University [2]NAVER AI Lab
{junhyuk.kim, jong-seok.lee}@yonsei.ac.kr bh.heo@navercorp.com

Abstract

Recently, learned image compression methods have outperformed traditional hand-crafted ones including BPG. One of the keys to this success is learned entropy models that estimate the probability distribution of the quantized latent representation. Like other vision tasks, most recent learned entropy models are based on convolutional neural networks (CNNs). However, CNNs have a limitation in modeling long-range dependencies due to their nature of local connectivity, which can be a significant bottleneck in image compression where reducing spatial redundancy is a key point. To overcome this issue, we propose a novel entropy model called Information Transformer (Informer) that exploits both global and local information in a content-dependent manner using an attention mechanism. Our experiments show that Informer improves rate–distortion performance over the state-of-the-art methods on the Kodak and Tecnick datasets without the quadratic computational complexity problem. Our source code is available at https://github.com/naver-ai/informer.

1. Introduction

More than one trillion photos are taken every year and the number is increasing [12], leading to an ever-increasing demand for improved compression efficiency. Recently, advances in deep learning have led to significant progress in learned image compression [7, 8, 16, 26, 32, 36, 40, 43–45]. Generally, learned image compression follows a transform coding framework [22] consisting of transformation and entropy coding (see the top of Fig. 1). In this framework, an image is first transformed into a quantized latent representation that enables more effective compression than the original image. Then, the quantized latent representation is encoded to a bitstream by a standard entropy coding algorithm (e.g., arithmetic coding [42]). An entropy model, i.e., a prior probability model on the quantized latent representation, is required for the entropy coding algorithm. Deep neural networks are employed for the transformation and entropy model in this framework [7, 8, 16, 32, 36, 40, 43], where both are learned in an end-to-end manner to fully utilize the strong capability of deep neural networks [25].

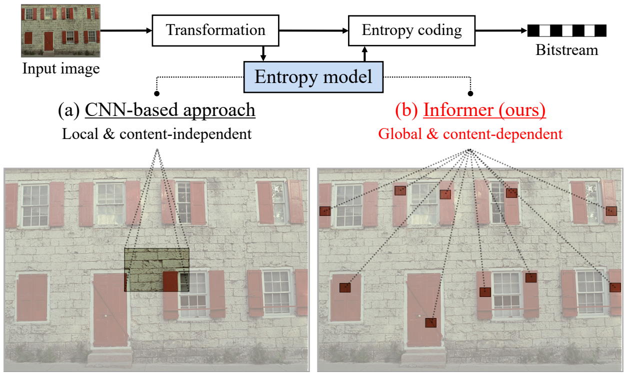


Figure 1. Overview of the proposed method. Our Informer is a learned entropy model capturing global dependencies in a content-dependent manner using the attention mechanism [46].

Since the length of the bitstream relies on the entropy model, designing an accurate entropy model is important for compression efficiency, which is our main focus in this paper. The goal of entropy models is to estimate a joint probability distribution over the elements of the quantized latent representation. A simple way to do this is to assume complete independence among the elements [7]. However, this approach yields limited compression efficiency since the assumption does not hold in most practical cases [8]. Thus, how to model the remaining dependencies has been an important issue in learned image compression [8,32,36,40]. It is popular to extract additional features, called "hyperprior" or "hierarchical prior", capturing the dependencies from the latent representation using convolutional neural networks (CNNs). This approach has contributed to learning accurate entropy models, making learned image compression methods outperform hand-crafted image codes such as BPG [10].

However, despite the significant progress, CNN-based entropy models still have limitations in capturing the dependencies due to the nature of CNNs. First, existing entropy models do not make full use of global information due to

*Work done while doing an internship at NAVER AI Lab.

the local receptive field of CNNs. This issue can be critical in modeling long-range dependencies. For example, in the case of Fig. 1, the CNN-based approach cannot fully capture the dependencies among the red windows that repeatedly appear across the whole image due to the localized receptive field. Second, the receptive fields of previous entropy models cannot exclude nearby elements with different contents due to the content-independent property of convolution operations [38]. In other words, no matter how different the contents of two elements are, they are processed within the same receptive field if they are located nearby. In Fig. 1, although the red window and the bricks have quite different contents, both are used simultaneously in the process of capturing dependencies.

To overcome these limitations, we propose a novel entropy model, called Information Transformer (Informer), that captures both global and local dependencies in a content-dependent manner using the attention mechanism of Transformer [46] (Fig. 1). In contrast to convolution operations, the attention mechanism has known to be effective in modeling long-range dependencies in a content-dependent manner [38]. Based on the joint autoregressive and hierarchical priors [36], which is the basis of the latest entropy models [16, 40], we introduce two novel hyperpriors, i.e., a global hyperprior and a local hyperprior. To model global dependencies of the quantized latent representation, our Informer first extracts a global hyperprior consisting of different vectors that attend to different areas of an image by using the cross-attention mechanism [4, 14, 34]. Furthermore, our Informer extracts a local hyperprior specialized for local information by using 1×1 convolutional layers. Our local hyperprior prevents our global hyperprior from utilizing only local information and thus allows our Informer to consider global and local information effectively.

Compared to the baseline entropy model [36], Informer improves the rate–distortion performance of learned image compression methods on the popular Kodak [31] and Tecnick [2] datasets. In addition, Informer achieves better performance than the recently proposed global reference model [40] aiming at capturing global dependencies; Informer not only yields higher rate–distortion performance but also avoids the quadratic computational complexity problem of the global reference model. Our main contributions can be summarized as follows:

- We propose joint global and local hyperpriors that effectively model two different types of dependencies between the elements of the quantized latent representation using the attention mechanism.
- We demonstrate that our Informer with the joint global and local hyperpriors improves rate–distortion performance of learned image compression while addressing the quadratic computational complexity problem.

2. Related work

Learned nonlinear transforms. One of the keys to the success of learned image compression is that deep neural networks effectively model nonlinear transforms suitable for image compression, while traditional image codecs mostly assume linear transforms due to the difficulty of hand-engineering nonlinear transforms for high-dimensional data like images [5]. Since Ballé *et al.* [6] proposed the generalized divisive normalization (GDN) layer that is effective for modeling nonlinear transforms, CNNs with the GDN layers have been widely used in later methods [7, 8, 28, 32, 36]. Recently, some learned nonlinear transforms have been proposed using deep residual networks with small kernels (i.e., 3×3) [15], an attention module [16], invertible neural networks [50], and an attentional multi-scale back-projection module [21].

Attention mechanisms. The attention mechanism [46] is one of the most successful methods to handle global information in deep neural networks. It demonstrates notable performance in the language domain through the Transformer architecture [46]. Some researches [29, 48, 49] tried to utilize the powerful performance of the attention mechanism in the computer vision domain. Recently, Vision Transformer [19] achieves state-of-the-art accuracy on image classification tasks. Many studies have been conducted to use and improve Vision Transformer in diverse vision tasks such as object detection [13, 24, 33], semantic segmentation [33, 52], and image quality assessment [17]. Since Transformer has a strong ability to model long-range dependencies regardless of their distance in the pixel domain [41], which existing learned entropy models do not have, we propose a novel Transformer-based learned entropy model.

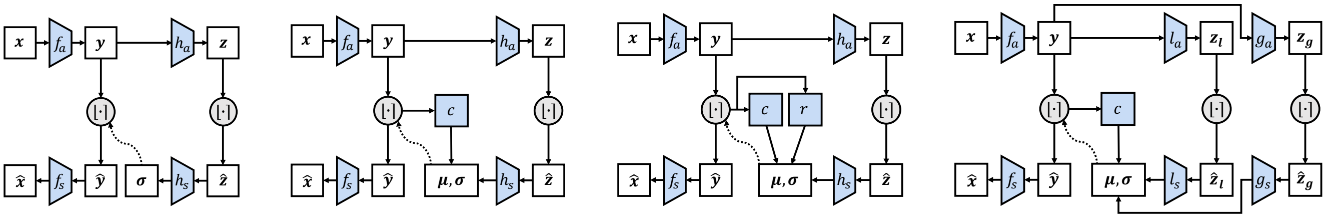
3. Joint global and local hyperpriors

3.1. Learned image compression

Given the input image $\boldsymbol{x}$, most learned image compression models [8, 16, 36] aim to jointly minimize the expected length of the bitstream (i.e., rate) and the expected distortion of the decoded image with respect to $\boldsymbol{x}$:

$$\underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}\left[-\log_2 p_{\hat{\boldsymbol{y}}}(\lfloor f_a(\boldsymbol{x})\rceil)\right]}_{\text{rate}}+\lambda\cdot\underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}\left[d(\boldsymbol{x},f_s(\lfloor f_a(\boldsymbol{x})\rceil))\right]}_{\text{distortion}}. \tag{1}$$

λ is the Lagrange multiplier controlling the trade-off between rate and distortion. $f_a(\cdot)$, $\lfloor\cdot\rceil$, and $f_s(\cdot)$ represent an encoder, a quantizer, and a decoder, respectively. $\hat{\boldsymbol{y}}$ is the quantized latent representation, i.e., $\hat{\boldsymbol{y}}=\lfloor f_a(\boldsymbol{x})\rceil$. $p_{\boldsymbol{x}}$ is the distribution of training images, and $p_{\hat{\boldsymbol{y}}}$ is the learned entropy model. When an entropy coding proceeds under the learned entropy model $p_{\hat{\boldsymbol{y}}}$, the smallest rate is the cross-entropy between the actual probability distribution of the quantized latent representation and the learned entropy model $p_{\hat{\boldsymbol{y}}}$. Thus,



(a) Hyperprior [8] (b) Context and hyperprior [36] (c) (b)+global reference [40] (d) Informer (ours)

Figure 2. Operational diagrams of learned image compression methods using different entropy models. The white blocks are data tensors, the blue blocks represent learned models, and the gray circles mean quantization operations.

the cross-entropy is used for the rate term. $d(\cdot,\cdot)$ in the distortion term is usually defined by traditional image distortion metrics such as mean squared error (MSE) or multi-scale structural similarity (MS-SSIM).

To enable gradient-based end-to-end training, studies have been conducted to deal with the non-differentiable quantization operation [1, 7, 43]. The most widely used method is to approximate quantization using additive uniform noise [7], which we adopt in this paper.

3.2. Learned entropy models

The entropy models seek to estimate a joint probability distribution over the elements of the quantized latent representation $\hat{\boldsymbol{y}}$. Note that the rate term in Eq. (1) is minimized when the learned entropy model $p_{\hat{\boldsymbol{y}}}$ perfectly matches the actual probability distribution. A simple approach to model the distribution of $\hat{\boldsymbol{y}}$ is to assume that all elements are statistically independent and to learn a fixed entropy model, i.e., fully factorized model [7, 43]. Despite its simplicity, this approach does not model the remaining dependencies in $\hat{\boldsymbol{y}}$, and thus cannot achieve optimal performance [8].

To address this limitation, advanced methods [8, 32, 36] propose conditional entropy models where the elements are assumed to follow conditionally independent parametric probability models, and the distribution parameters are adapted by utilizing the remaining dependencies. They can be divided into two directions: 1) what parametric models to be used [8, 16, 18, 36] and 2) how to model dependencies [8, 32, 36, 40]. The former direction includes zero-mean Gaussian [8], Gaussian [36], Gaussian mixture [16], and asymmetric Gaussian [18]. Among them, we employ the widely used one, i.e., Gaussian [36]. Specifically, we use a Gaussian distribution convolved with a unit uniform distribution following the previous works [8, 36]:

$$p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}}) = \prod_i \Big(\mathcal{N}(\mu_i, \sigma_i^2) * \mathcal{U}(-\tfrac{1}{2}, \tfrac{1}{2})\Big)(\hat{y}_i), \qquad (2)$$

where μ_i and σ_i are the mean and scale parameters of the Gaussian distribution for each element $\hat{y}_i$, respectively.

The main focus of this study is on the latter direction, i.e., accurate modeling of dependencies. Existing methods model local dependencies in two different ways. First, Ballé *et al.* [8] capture the local dependencies by extracting side information that is encoded additionally, which is called a hyperprior. The operational diagram of this approach is explained in Fig. 2a. The hyperprior model (h_a and h_s) extracts and utilizes the hyperprior $\hat{\boldsymbol{z}}$ for predicting the distribution parameter $\boldsymbol{\sigma}$. Since additional information is encoded, the rate term in Eq. (1) is extended as follows:

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}\big[-\log_2 p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}}) - \log_2 p_{\hat{\boldsymbol{z}}}(\hat{\boldsymbol{z}})\big], \qquad (3)$$

where the learned entropy model $p_{\hat{\boldsymbol{z}}}$ is designed using the non-parametric fully factorized entropy model [7].

Another approach of modeling the local dependencies is to utilize previously decoded adjacent elements (i.e., a context prior [32, 36]). While the hyperprior requires additional bits, the context prior is bit-free. Since Minnen *et al.* [36] and Lee *et al.* [32] demonstrated that the two kinds of priors are complementary, they have been typically used jointly in literature [16, 18, 21] (Fig. 2b). The outputs of the context model c and the hyperprior model h_a and h_s are used together for predicting the distribution parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$.

While the above approaches focus on modeling the local dependencies, Qian *et al.* [40] propose a global reference model that captures long-range dependencies. This utilizes the most relevant previously decoded element for estimating distribution parameters of the current element. As shown in Fig. 2c, Qian *et al.* [40] use the global reference model r combined with the joint context and hyperprior model [36].

Motivation of Informer. Although utilizing the global dependencies is an innovative direction, the global reference model [40] does not fully exploit global information because only a single element among previously decoded ones is used. To improve the utilization of global information, we introduce a global hyperprior $\hat{\boldsymbol{z}}_g$ extracted from all elements of the latent representation $\boldsymbol{y}$ as shown in Fig. 2d.

In addition, the global reference model [40] has an issue that the computational complexity increases quadratically with respect to the given image size. This is because in its self-attention-like mechanism, the most similar element to the current one is searched among all the previously decoded elements and this process is repeated for all elements. In order to avoid such an issue, in our global hyperprior modeling, we utilize a cross-attention mechanism with a fixed number of query regardless of the image size.

3.3. Decomposition of hyperpriors

As shown in Fig. 2d, extending the context and hyperprior [36], our entropy model, Informer, decomposes the hyperprior into two novel hyperpriors: a global hyperprior $\hat{\boldsymbol{z}}_g$ and a local hyperprior $\hat{\boldsymbol{z}}_l$. Fig. 3 illustrates a high-level overview of our hyperpriors in comparison with the previous approach [8, 36]. The hyperprior $\hat{\boldsymbol{z}}$ in the previous approach reduces the spatial resolution of the latent representation $\boldsymbol{y}$ while retaining the number of channels. Due to the localized operations in CNNs, it uses limited local information containing only spatially adjacent elements.

On the other hand, our global hyperprior $\hat{\boldsymbol{z}}_g$ consists of vectors having no spatial information and is not limited to the local area. Thus, it can handle the whole image area when modeling dependencies. Specifically, the global dependencies are modeled in a content-dependent manner by using the attention mechanism [46]. In addition, the local hyperprior $\hat{\boldsymbol{z}}_l$ models inter-channel dependencies in each spatial location to complement the lack of spatial components in $\hat{\boldsymbol{z}}_g$. It maintains the spatial resolution of the latent representation $\boldsymbol{y}$ while reducing the number of channels. In summary, the proposed two types of hyperpriors are extracted in parallel capturing dependencies of the latent representation $\boldsymbol{y}$ by effectively complementing each other.

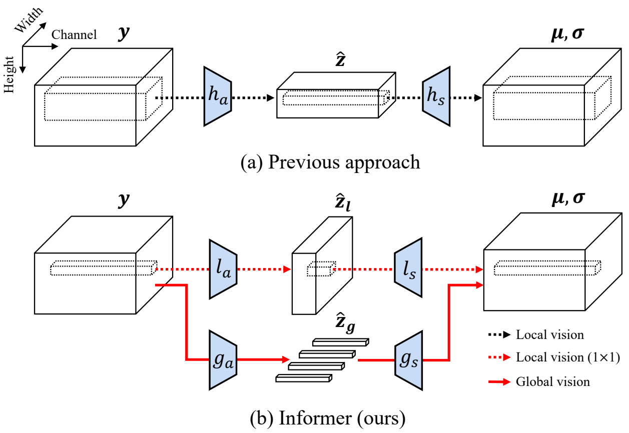
To extract and utilize the hyperpriors, we build *Global Hyperprior Model* (i.e., *Global Hyper Encoder* g_a and *Global Hyper Decoder* g_s) and *Local Hyperprior Model* (i.e., *Local Hyper Encoder* l_a and *Local Hyper Decoder* l_s). All trainable models (the blue blocks in Fig. 2d) are learned using Eq. (1) with extending the rate term in order to consider our hyperpriors $\hat{\boldsymbol{z}}_l$ and $\hat{\boldsymbol{z}}_g$:

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}\big[-\log_2 p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}}) - \log_2 p_{\hat{\boldsymbol{z}}_l}(\hat{\boldsymbol{z}}_l) - \log_2 p_{\hat{\boldsymbol{z}}_g}(\hat{\boldsymbol{z}}_g)\big], \tag{4}$$

where the learned entropy models $p_{\hat{\boldsymbol{z}}_l}$ and $p_{\hat{\boldsymbol{z}}_g}$ are designed using the non-parametric fully factorized entropy model [7].

3.4. Modeling of hyperpriors

Global hyperprior. Given the input latent representation $\boldsymbol{y} \in \mathbb{R}^{H\times W\times C}$, where H, W, and C are the height, width, and the number of channels, respectively, the process of extracting the global hyperprior $\hat{\boldsymbol{z}}_g$ is illustrated in *Global Hyper Encoder* in Fig. 4. We define fixed-size global tokens $\boldsymbol{u} \in \mathbb{R}^{N\times C}$ as query of the multi-head attention layer of *Global Hyper Encoder*. N is a predefined parameter, which is fixed to eight in our final models. Note that the global tokens are learnable parameters that are determined through end-to-end training like other network parameters. With the multi-head attention block $MHA(q,k,v)$, the MLP block $MLP_1(\cdot)$, and the last MLP layer $MLP_2(\cdot)$, the formulation of our *Global Hyper Encoder* is as follows:



(a) Previous approach

(b) Informer (ours)

Figure 3. Schematic illustration of a typical hyperprior and the proposed hyperpriors. In contrast to the hyperprior based on spatial dimension reduction [8,36], our Informer utilizes a global hyperprior $\hat{\boldsymbol{z}}_g$ extracted by an attention mechanism and a local hyperprior $\hat{\boldsymbol{z}}_l$ specialized for spatial information. Quantization operations after h_a, l_a, and g_a are omitted for simplicity.

$$\begin{aligned} \boldsymbol{u}' &= MHA(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{y}), \\ \boldsymbol{z}_g &= MLP_2(MLP_1(\boldsymbol{u}')), \end{aligned} \tag{5}$$

where the normalization layers [3] are omitted for simplicity. $MHA(q,k,v)$ models global dependencies. $MLP_1(\cdot)$ and $MLP_2(\cdot)$ extract $\boldsymbol{z}_g \in \mathbb{R}^{N\times \frac{C}{N}}$ for further modeling of inter-channel dependencies. After the quantization operation $\lfloor\cdot\rceil$, the global hyperprior $\hat{\boldsymbol{z}}_g$ is obtained. *Global Hyper Decoder*, which is one linear layer, receives the global hyper prior $\hat{\boldsymbol{z}}_g$ as the input and generates $\boldsymbol{\psi}_g \in \mathbb{R}^{N\times 2C}$.

Local hyperprior. In order to model inter-channel dependencies at each spatial location, we design *Local Hyperprior Model* by stacking 1×1 convolutional layers with the leaky ReLU activation [35], which is shown in the right part of Fig. 4. From the input $\boldsymbol{y}$, *Local Hyper Encoder* extracts the local hyperprior $\hat{\boldsymbol{z}}_l \in \mathbb{R}^{H\times W\times \frac{C}{16}}$. *Local Hyper Decoder* utilizes $\hat{\boldsymbol{z}}_l$ and yields its output $\boldsymbol{\psi}_l \in \mathbb{R}^{H\times W\times 2C}$.

3.5. Prediction of distribution parameters

As shown in Fig. 2d, we also employ *Context Model* c that uses the previously decoded elements $\hat{\boldsymbol{y}}_{<i}$. We adopt the same structure as in Minnen *et al.* [36], i.e., one 5×5 masked convolutional layer. For predicting the distribution parameters, i.e., mean $\boldsymbol{\mu} \in \mathbb{R}^{H\times W\times C}$ and scale $\boldsymbol{\sigma} \in \mathbb{R}^{H\times W\times C}$, we combine the outputs from *Context Model*, *Local Hyperprior Model*, and *Global Hyperprior Model*. Since the output of *Global Hyperprior Model* $\boldsymbol{\psi}_g \in \mathbb{R}^{N\times 2C}$ does not have the spatial dimension, conventional feature

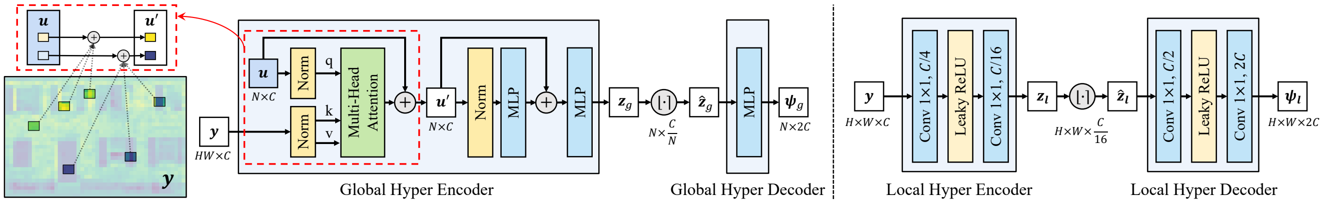


Figure 4. Structure of *Global Hyperprior Model* and *Local Hyperprior Model* in our Informer. The left part illustrates a multi-head attention-based *Global Hyperprior Model*. It utilizes global tokens $\boldsymbol{u} \in \mathbb{R}^{N \times C}$ to extract fixed-size (i.e., $N \times C$) global information from images having various sizes. In the dotted box, we visualize the process of the attention mechanism for the case of $N = 2$, where the channel dimension is omitted for simplicity. The global tokens $\boldsymbol{u}$, which are image-independent universal features, are converted to image-dependent features $\boldsymbol{u}'$ by attending the content of the input $\boldsymbol{y}$. The right part shows a *Local Hyperprior Model* consisting of 1×1 convolutional layers with strides of one that preserve spatial information of images. For the convolutional layers, the number of kernels are specified.

combining methods such as concatenation or element-wise addition are inapplicable. Therefore, we propose a new *Parameter Model* with a multi-head attention-based combining method, which is shown in Fig. 5. The outputs of *Context Model* and *Global Hyperprior Model* ($\boldsymbol{\phi}$ and $\boldsymbol{\psi}_g$, respectively) are combined by using the multi-head attention block. After the multi-head attention block and following MLP block, the result is concatenated with the output of *Local Hyperprior Model* $\boldsymbol{\psi}_l$ and the distribution parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are generated via three 1×1 convolutional layers. The leaky ReLU activation functions are used after the first two convolutional layers, which are omitted in Fig. 5. The formulation of *Parameter Model* is as follows:

$$\begin{aligned} &\{\boldsymbol{\mu}, \boldsymbol{\sigma}\} = pm(\boldsymbol{\phi}, \boldsymbol{\psi}_g, \boldsymbol{\psi}_l) \\ &\text{with } \boldsymbol{\phi} = c(\hat{\boldsymbol{y}}_{<i}),\ \boldsymbol{\psi}_g = gh(\boldsymbol{y}),\ \text{and } \boldsymbol{\psi}_l = lh(\boldsymbol{y}), \end{aligned} \tag{6}$$

where $pm(\cdot)$, $c(\cdot)$, $gh(\cdot)$, and $lh(\cdot)$ represent *Parameter Model*, *Context Model*, *Global Hyperprior Model*, and *Local Hyperprior Model*, respectively.

4. Experiments

All experiments are conducted on a PyTorch [39] based open-source library: CompressAI platform [9], which has recently been introduced for developing and evaluating learning-based image codecs.

Training. Our models are trained with various configurations of the Lagrange multiplier λ and the distortion metric $d(\cdot,\cdot)$. We use MSE and $(1 - \text{MS-SSIM})$ for $d(\cdot,\cdot)$, and set $\lambda \in \{0.0018, 0.0035, 0.0067, 0.0130, 0.0250, 0.0483\}$ for MSE and $\lambda \in \{2.40, 4.58, 8.73, 16.64, 31.73, 60.50\}$ for MS-SSIM; thus, a total of 12 final models are obtained. We use the training images of the Triplet dataset, which is a subset of Vimeo-90K [51]. We use a batch size of 8 with 256×256 patches randomly cropped from the training images. All models are trained using the Adam optimizer [30] for 250 epochs. The learning rate starts at 10^{-4} and decreases to one-third at the 150th, 180th, 210th, and 240th epochs. For the MS-SSIM-optimized models, we finetune the MSE-optimized models with a learning rate of 10^{-5}.

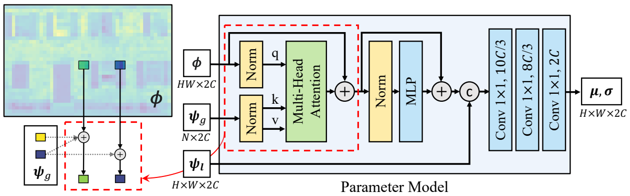
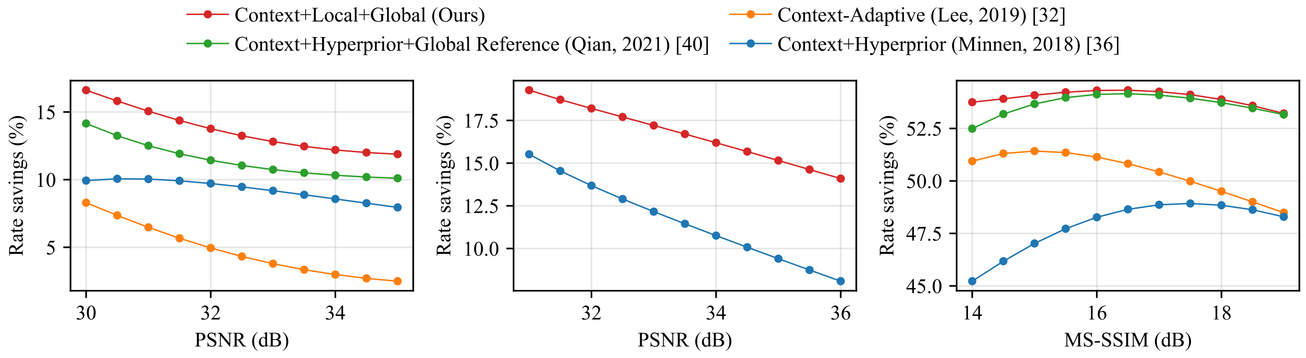


Figure 5. Structure of *Parameter Model*. It combines the outputs of *Context Model*, *Global Hyperprior Model*, and *Local Hyperprior Model* ($\boldsymbol{\phi}$, $\boldsymbol{\psi}_g$, and $\boldsymbol{\psi}_l$, respectively), and predicts the distribution parameters (mean $\boldsymbol{\mu}$ and scale $\boldsymbol{\sigma}$). In the dotted box, we visualize the process of the attention mechanism ($N = 2$), where the channel dimension is omitted for simplicity. The local location-specific information $\boldsymbol{\phi} \in \mathbb{R}^{H \times W \times 2C}$ is updated by attending global location-free information $\boldsymbol{\psi}_g \in \mathbb{R}^{N \times 2C}$. For the convolutional layers, the number of kernels are specified.

Evaluation. We evaluate our method on the Kodak [31] and Tecnick [2] datasets, which are commonly used for benchmarking learned image compression methods. The Kodak dataset consists of 24 images with a resolution of 768×512 pixels. The Tecnick dataset includes 100 images with a resolution of 1200×1200 pixels. At the encoding stage, we use the asymmetric numeral systems [20] for the entropy coding. To evaluate rate–distortion performance, we use the bits per pixel (bpp) and either the peak signal-to-noise ratio (PSNR) or MS-SSIM depending on the distortion metric $d(\cdot,\cdot)$. MS-SSIM is converted into decibels, i.e., $-10\log_{10}(1 - \text{MS-SSIM})$ [8].

4.1. Performance comparison

Comparison with other entropy models. To show the effectiveness of Informer, we compare its rate–distortion performance with that of the state-of-the-art image compression methods [32, 36, 40] in which the transformation parts are modeled similar to that of Minnen *et al.* [36] and different entropy models are used. Each of their encoder and decoder consists of four convolutional layers. Except for the method of Qian *et al.* [40], which uses its own generalized subtractive and divisive normalization (GSDN), all methods



(a) (b) (c)

Figure 6. Performance of image compression methods using different entropy models. Each curve represents the rate savings (%) relative to BPG [10] at different quality levels. Larger values mean better performance. The results of the MSE-optimized methods are averaged over (a) Kodak [31] and (b) Tecnick [2], respectively, and (c) the results of MS-SSIM-optimized methods are averaged on Kodak [31].

Method	320×240	480×360	640×480	768×512	1280×720	1920×1080	4096×2304
Context+Hyperprior [36]	2.73	6.14	10.91	13.96	32.73	73.64	335.13
Context+Hyperprior+Global Reference [40]	9.98	22.85	41.59	54.03	138.04	366.57	3296.99
Informer (ours)	**2.69**	**6.04**	**10.73**	**13.73**	**32.15**	**72.34**	**329.17**

Table 1. Comparison of complexity of different entropy models at various image sizes. Each value means GFLOPs of each entropy model when the image having the corresponding resolution is used for entropy modeling.

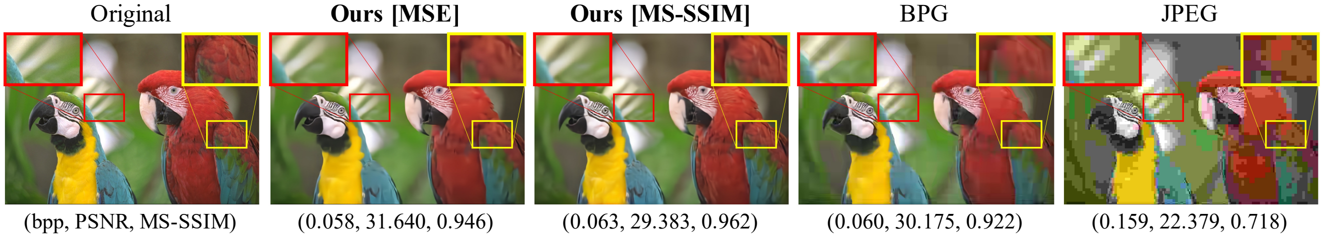


Figure 7. Visual comparison of the decoded images by our methods and the other image codecs on “Kodim23” from Kodak [31].

use GDN [6] as activation functions. But, for a fair comparison, we use the values of the variant of Qian *et al.* [40] using GDN, which is reported in their paper. Because Qian *et al.* [40] do not report the performance of the variant optimized with MS-SSIM, we use the values of the MS-SSIM-optimized full models using GSDN in their paper.

Fig. 6 shows the relative rate savings compared to BPG [10] at different quality levels. Note that these graphs are generalized versions of the popular Bjøntegaard Delta (BD) chart [11] used for evaluation of image compression. As shown in Figs. 6a and b, our MSE-optimized models show significant rate savings over BPG, ranging from 11.84% to 19.27%. Compared to the baseline entropy model, i.e., “Context+Hyperprior (Minnen, 2018) [36]”, Informer significantly improves rate savings across all PSNR levels on both datasets. This shows that our global and local hyperpriors improve the accuracy of the learned entropy model. Even compared to the entropy model aiming at modeling global dependencies, i.e., “Context+Hyperprior+Global Reference (Qian, 2021) [40]”, Informer shows superiority across all PSNR levels on Kodak [31] with performance gaps from 1.74% up to 2.60 %. These results demonstrate the effectiveness of Informer’s capability to model joint global and local dependencies. Furthermore, when optimized with MS-SSIM (Fig. 6c), Informer outperforms all the other models on Kodak [31], which achieves rate savings by large margins up to 54.32%.

Complexity. We evaluate the performance of our Informer in terms of computational efficiency. For this, we calculate the number of floating-point operations (FLOPs) required for various image sizes. Tab. 1 shows GFLOPs for our Informer and other entropy models [36, 40]. Informer shows the best performance across all image sizes. In particular, Informer does not suffer from the quadratic computational complexity problem that the “Con-

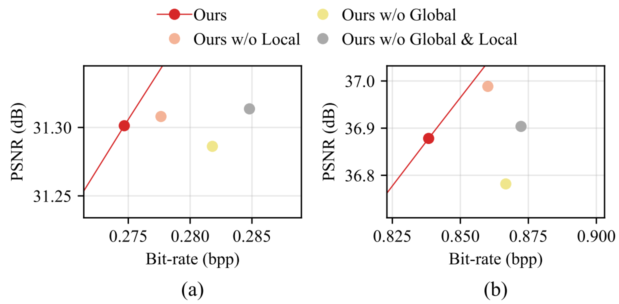


Figure 8. Ablation studies of our global and local hyperpriors in Informer. (a) $\lambda = 0.0067$ and (b) $\lambda = 0.0483$.

text+Hyperprior+Global Reference [40]” model has. In other words, as the image resolution increases, GFLOPs of “Context+Hyperprior+Global Reference [40]” increases quadratically, being 10 times larger at the resolution of 4096×2304 than at the resolution of 1920×1080, while the complexity of Informer has a near-linear scale. This is because Informer models global dependencies using a cross-attention mechanism with a fixed number of query.

Qualitative performance. We compare visual quality of our decoded images with those of other traditional image codecs on the Kodak dataset [31]. For the image codecs, we employ JPEG [47] and BPG [10]. For a fair comparison, we encode the images under compression settings for as similar bpp values as possible. Fig. 7 shows that our method produces clear patterns (red boxes) and does not lose details (yellow boxes) compared to the other results.

4.2. Model analysis

We perform further analysis of Informer. For this, we utilize additional MSE-optimized models in two different bpp regions using $\lambda = 0.0067$ and $\lambda = 0.0483$, respectively.

Ablation study. To evaluate the contribution of the proposed hyperpriors in Informer, we conduct ablation studies. The rate–distortion performance with or without each hyperprior is shown in Fig. 8. It is shown that the original Informer shows the best results regardless of the bpp region. Introducing our global hyperprior leads to significant improvement compared to the entropy model using only the context prior (i.e., “Ours w/o Global & Local”), while our local hyperprior seems to be more helpful when combined with our global hyperprior than when used alone.

Analysis of decomposition. To validate the effectiveness of our approach to decomposing hyperpriors, we compare two different decomposition methods in Fig. 9: Informer (“Context+Global+Local”) and the existing hyperprior with our global hyperprior (“Context+Hyperprior+Global”). As a reference, we also provide the results of the existing context and hyperprior model [36] (“Context+Hyperprior”), which uses a single hyperprior without decomposition. We

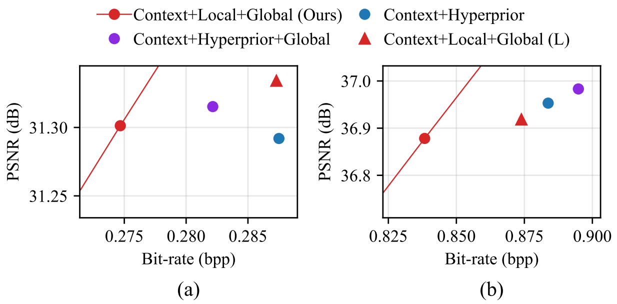


Figure 9. Comparison of different decomposition methods (circles with different colors) and different combining methods (red markers with different shapes). (a) $\lambda = 0.0067$ and (b) $\lambda = 0.0483$.

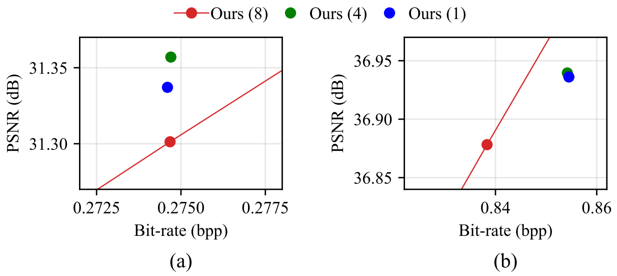


Figure 10. Model analysis with respect to the number of global tokens N, which is indicated by the number in parentheses. (a) $\lambda = 0.0067$ and (b) $\lambda = 0.0483$.

observe that Informer is better than the others for both bpp regions. The “Context+Hyperprior+Global” method shows even similar or slightly worse rate–distortion performance than the baseline (“Context+Hyperprior”) in the higher bpp region. We conjecture that this is due to the overlap in the roles of the existing hyperprior and our global hyperprior. Since the hyperprior-based approach requires additional bit allocation, redundant roles can be an obstacle for improving performance due to excessive increase of bit usage, which implies that elaborate modeling of hyperpriors is important.

Analysis of combining methods. In Fig. 9, Informer using a variant of *Parameter Model*, denoted as “Context+Local+Global (L)”, is also evaluated, which combines the inputs to *Parameter Model* in a different way. Specifically, the output of *Local Hyperprior Model* ψ_l is used as query for the attention layer, and the output of *Context Model* ϕ is concatenated after the MLP block. Our method shows better performance than this variant. In other words, the global hyperprior is more effective when used for updating the context prior than updating the local hyperprior.

Analysis of attention. We examine the effect of the number of global tokens N on rate–distortion performance in Fig. 10. We observe that using four global tokens (“Ours (4)”) yields the best performance at the lower bpp region, while the best performance at the higher bpp region is ob-

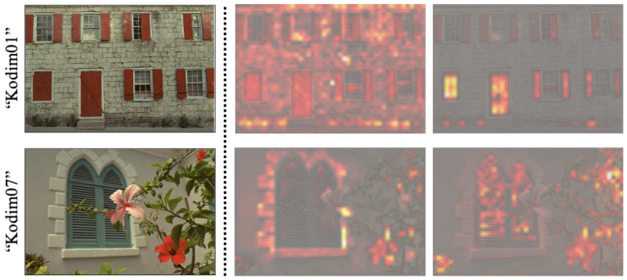


Figure 11. Visualization of the attention map used by the attention mechanism of the proposed *Global Hyper Encoder*. Two sample images from Kodak [31] are used, and two attention maps corresponding to two example global tokens are shown.

tained using eight global tokens (“Ours (8)”). This is probably due to the different degrees of remaining dependencies between the quantized latent representation $\hat{\boldsymbol{y}}$. In other words, a large number of global tokens is effective to capture detailed dependencies for the higher bpp region.

In addition, Fig. 11 visualizes attention maps used by the attention mechanism of our *Global Hyper Encoder*, which show where the example global tokens pay attention to for capturing global dependencies. The results show that the global tokens successfully capture global information over the whole image area and different global tokens utilize different image regions in a content-dependent manner. For example, in the case of “kodim07”, the global token shown on the left side attends to the bricks located along the edge of the window, while the global token shown on the right side attends to the window.

4.3. Global hyperprior vs. global context prior

To examine whether the superior performance of Informer originates from the strong capability of the attention mechanism-based structure or introduction of the joint global and local hyperpriors, we evaluate another entropy model aiming at capturing global dependencies. Building on the context and hyperprior entropy model [36] as in Informer, we newly design *Global Context Model* using the masked multi-head self-attention mechanism [46]. As shown in Fig. 12a, *Global Context Model* receives the output of *Context Model* ϕ and updates it by attending the previously decoded elements. Through the process of the attention mechanism, *Global Context Model* considers multiple references, and thus it can be seen as a generalized version of the global reference model [40] utilizing only the most relevant reference. Tab. 2 shows that Informer is superior to the method using *Global Context Model*. We argue that introducing the self-attention mechanism is not a silver bullet for image compression, but careful design consideration should be accompanied with it.

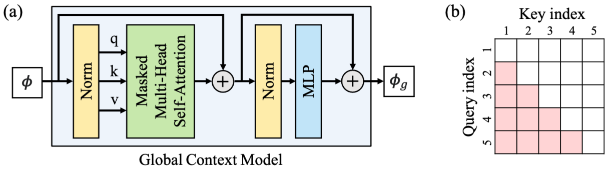


Figure 12. Global context model. (a) Structure and (b) an example of masked self-attention map where the red boxes are activated.

λ	Method	Rate (bpp)	PSNR (dB)
0.0067	Informer	**0.2763**	**31.3254**
	Global context	0.2842	31.2947
0.0483	Informer	**0.8514**	**36.9739**
	Global context	0.8718	36.9134

Table 2. Comparison of the different methods of introducing global vision to learned entropy models. The “Global context” model introduces global vision to *Context Model* using the masked self-attention mechanism. The results are obtained at 160 epochs.

5. Conclusion

We proposed a novel learned entropy model (Informer) for learned image compression. Based on the previous joint autoregressive and hierarchical priors [36], Informer introduced two different hyperpriors for modeling remaining dependencies in the quantized latent representation, one for global dependencies and the other for local dependencies. We showed that Informer outperforms existing entropy models in terms of rate–distortion performance with computational efficiency. Beyond the existing CNN-based localized dependency modeling methods, Informer presents a completely new approach that effectively utilizes both global and local information in a content-dependent manner using the attention mechanism.

6. Limitation

Informer cannot be parallelized in its decoding process, which is due to the inherent limitation of the autoregressive prior that utilizes only previously decoded elements. We expect that this problem can be addressed by combining our approach with the channel-wise autoregressive model [37] or bidirectional context model [23].

Acknowledgement

All experiments were conducted on the NAVER Smart Machine Learning (NSML) platform [27]. This work was supported by the NRF grant funded by the Korea government (MSIT) (2021R1A2C2011474).

References

[1] Eirikur Agustsson and Lucas Theis. Universally quantized neural compression. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[2] Nicola Asuni and Andrea Giachetti. TESTIMAGES: a large-scale archive for testing visual devices and basic image processing algorithms. In *Proceedings of the Smart Tools and Apps for Graphics (STAG)*, 2014. 2, 5, 6

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[4] Song Bai, Philip Torr, et al. Visual Parser: Representing part-whole hierarchies with transformers. *arXiv preprint arXiv:2107.05790*, 2021. 2

[5] Johannes Ballé, Philip A Chou, David Minnen, Saurabh Singh, Nick Johnston, Eirikur Agustsson, Sung Jin Hwang, and George Toderici. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020. 2

[6] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. Density modeling of images using a generalized normalization transformation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. 2, 6

[7] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 1, 2, 3, 4

[8] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 3, 4, 5

[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. 5

[10] Fabrice Bellard. BPG image format. http://bellard.org/bpg/. 1, 6, 7

[11] Gisle Bjøntegaard. Calculation of average PSNR differences between RD-curves. *VCEG-M33*, 2001. 6

[12] 2021 Worldwide Image Capture Forecast: 2020 – 2025. https://riseaboveresearch.com/rar-reports/2021-worldwide-image-capture-forecast-2020-2025/. 1

[13] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[14] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[15] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Deep residual learning for image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2019. 2

[16] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3

[17] Manri Cheon, Sung-Jun Yoon, Byungyeon Kang, and Junwoo Lee. Perceptual image quality assessment with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2021. 2

[18] Ze Cui, Jing Wang, Shangyin Gao, Tiansheng Guo, Yihui Feng, and Bo Bai. Asymmetric gained deep image compression with continuous rate adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 2

[20] Jarek Duda. Asymmetric numeral systems: entropy coding combining speed of Huffman coding with compression rate of arithmetic coding. *arXiv preprint arXiv:1311.2540*, 2013. 5

[21] Ge Gao, Pei You, Rong Pan, Shunyuan Han, Yuanyuan Zhang, Yuchao Dai, and Hojae Lee. Neural image compression via attentional multi-scale back projection and frequency decomposition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[22] Vivek K Goyal. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, 18(5):9–21, 2001. 1

[23] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8

[24] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[25] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. 1

[26] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[27] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. NSML: Meet the MLaaS platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. 8

[28] Jun-Hyuk Kim, Jun-Ho Choi, Jaehyuk Chang, and Jong-Seok Lee. Efficient deep learning-based lossy image compression via asymmetric autoencoder and pruning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. 2

[29] Jun-Hyuk Kim, Jun-Ho Choi, Manri Cheon, and Jong-Seok Lee. MAMNet: Multi-path adaptive modulation network for image super-resolution. *Neurocomputing*, 402:38–49, 2020. 2

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[31] The Kodak PhotoCD dataset. http://r0k.us/graphics/kodak/. 2, 5, 6, 7, 8

[32] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 1, 2, 3, 5

[33] 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 International Conference on Computer Vision (ICCV)*, 2021. 2

[34] Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. Luna: Linear unified nested attention. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[35] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013. 4

[36] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[37] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2020. 8

[38] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 5

[40] Yichen Qian, Zhiyu Tan, Xiuyu Sun, Ming Lin, Dongyang Li, Zhenhong Sun, Li Hao, and Rong Jin. Learning accurate entropy model with global reference for image compression. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 3, 5, 6, 7, 8

[41] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[42] Jorma Rissanen and Glen Langdon. Universal modeling and coding. *IEEE Transactions on Information Theory*, 27(1):12–23, 1981. 1

[43] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 1, 3

[44] George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. 1

[45] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2, 4, 8

[47] Gregory K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. 7

[48] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Standalone axial-attention for panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[49] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[50] Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. Enhanced invertible encoding for learned image compression. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2021. 2

[51] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019. 5

[52] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2