GLIC: General Format Learned Image Compression

MingSheng Zhou, MingMing Kong*

School of Computer and Software Engineering, XiHua University, ChengDu, SiChuan, China, 610039
mingshengzhou@foxmail.com, kongming000@126.com

Abstract

Learned image lossy compression techniques have surpassed traditional methods in both subjective vision and quantitative evaluation. However, current models are only applicable to three-channel image formats, limiting their practical application due to the diversity and complexity of image formats. We propose a high-performance learned image compression model for general image formats. We first introduce a transfer method to unify any-channel image formats, enhancing the applicability of neural networks. This method's effectiveness is demonstrated through image information entropy and image homomorphism theory. Then, we introduce an adaptive attention residual block into the entropy model to give it better generalization ability. Meanwhile, we propose an evenly grouped cross-channel context module for progressive preview image decoding. Experimental results demonstrate that our method achieves state-of-the-art (SOTA) in the field of learned image compression in terms of PSNR and MS-SSIM. This work extends the applicability of learned image compression techniques to more practical production environments.

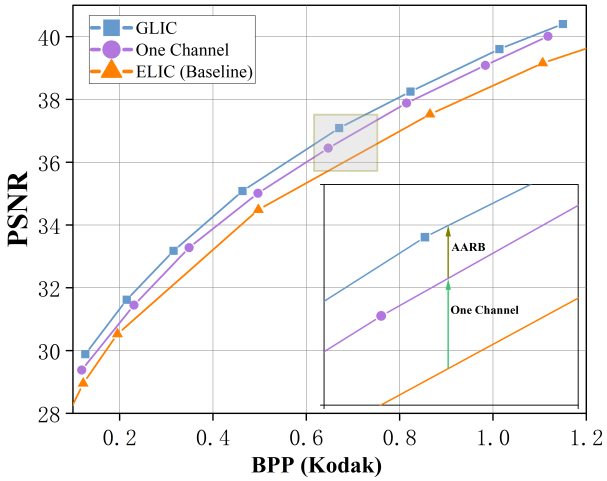


Figure 1: Improvement in RD performance by the one-channel and AARB schemes compared to the baseline model (He et al. 2022).

Introduction

In recent years, image compression technology has advanced rapidly. Subjective visual and quantified indicators such as PSNR (Gonzales and Wintz 1987) and MS-SSIM (Wang, Simoncelli, and Bovik 2003) have surpassed existing hand-crafted algorithms like JPEG (Pennebaker and Mitchell 1992) and VTM (Wien and Bross 2020), etc. Learned image compression also outperforms these algorithms in terms of rate-distortion (RD) performance at the same decoding quality. On one hand, traditional hand-crafted image compression algorithms typically support three-channel (RGB), one-channel (grayscale), or four-channel (RGBA) images, etc. On the other hand, existing learned image compression methods (Ballé et al. 2018; Cheng et al. 2020; He et al. 2021, 2022; Gao et al. 2022; Xu et al. 2022; Lieberman et al. 2023; Lee, Jeong, and Kim 2022; Ali et al. 2024) only model quantization for three-channel images and cannot be extended to multi-channel images in general format, such as grayscale images or medical tomography images. Expanding the model's applicability to general image format compression and progressive decoding is necessary for the practical application of learned image compression techniques.

Significant advancements have occurred in learned image compression, particularly regarding encoding and decoding performance. For example, (He et al. 2021) introduced a novel context encoding scheme based on a checkerboard, which enabled parallel computation in autoregressive context encoding and significantly enhanced encoding efficiency. (Minnen and Singh 2020) proposed a channel context coding scheme with even grouping, substantially improving the decoding process's efficiency. (He et al. 2022) presented an unevenly grouped channel context encoding scheme, further improving decoding efficiency and introducing a new preview image decoding scheme. Despite these advancements, extending the learned image compression technique to a general image format for better RD performance remains challenging. Our research is driven by the need to use this technique to process any image format and improve RD performance without compromising en/decode speed.

*Corresponding author.

We find that by converting the image (regardless of the number of channels) into a one-channel two-dimensional matrix and combining it with an adaptive attention residual block (AARB). It can effectively avoid the negative effects caused by channel correlation and improve the homogeneity index of the potential representation, which in turn improves the RD performance, as Figure 1.

In this paper, we contribute to this learned image compression field in the following ways:

- We propose a General Format Learned Image Compression (GLIC) model. It can unify any image format into a general format for general image compression. It obtains better RD performance than existing models across multiple datasets.
- We improve an Adaptive Attention Residual Block (AARB). It extracts the high-level and overall semantic information of an image separately, and obtains their weights adaptively to make the model more applicable.
- We propose an evenly grouped Cross-Channel Context (CCCT) module. It priority analyzes the overall information of the image, and then gradually decodes the complete image information to achieve fast and high-quality progressive image decoding.

Related Works

Traditional hand-crafted image compression algorithms are vital for Internet applications. However, with the growing demand for massive data storage across various industries, these traditional methods struggle to meet future needs. Researchers are exploring ways to improve image compression efficiency using neural network techniques. One of the earliest techniques combines the Variational Auto Encoding model (VAE) with the Image Entropy Coding, proving the feasibility of neural network in image compression. The subsequent introduction of the Hyperprior model, the Autoregressive context module, and the Parallel context module has further improved the rate-distortion performance and co/decode speed of learned image compression, making it more suitable for practical applications.

Basic model

Learned lossy image compression (Ballé et al. 2018; Cheng et al. 2020; He et al. 2021, 2022; Mentzer et al. 2018; Minnen, Ballé, and Toderici 2018; Minnen and Singh 2020; Ballé, Laparra, and Simoncelli 2016; Yang and Mandt 2024; Li et al. 2024; Liu, Sun, and Katto 2023) is an end-to-end optimization technique that combines transform coding (Goyal 2001; Ballé et al. 2020) and entropy coding techniques (Rissanen and Langdon 1981; Martin 1979; Van Leeuwen 1976), its optimization objective requires a trade-off between rate-distortion and image distortion:

$$\begin{aligned}\mathbb{L} = R(\hat{y}) + \lambda \cdot D(x, g_s(\hat{y})) = \\ \mathbb{E}_{x\sim p_x}[-\log_2 p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z}) - \log_2 p_{\hat{z}}(\hat{z})] \\ + \lambda \cdot \mathbb{E}_{x\sim p_x}[d(x,\hat{x})] \quad (1)\end{aligned}$$

where R is the rate term, D is the distortion term, and λ is the Lagrange Multiplier Hyperparameter, which is used to control the distortion trade-off. As Figure 2a, $x = \{x_1, x_2, x_3\}$ is the original image, x_1, x_2 and x_3 are the three channels of a RGB image. $\hat{x} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3\}$ is the decoded image. y is the latent representation before quantization, and $\hat{y}$ are discrete symbols that need to be persistently saved after entropy encoding. $\hat{y} = \lceil g_a(x) \rfloor$ is a quantization of the latent representation obtained by the image analysis transform $y = g_a(x)$, $\lceil \cdot \rfloor$ is a quantitative operation. $\hat{x} = g_s(\hat{y})$ indicates that a decoded image is obtained by image synthesis transform from the quantization result. $U|Q$ are quantization and entropy coding operations. During training, quantization is emulated using uniform noise $U\left(-\frac{1}{2}, \frac{1}{2}\right)$ to produce noisy codes $\tilde{y}$. In the en/decode stage, $U|Q$ denotes the actual round quantization responsible for generating $\hat{y}$.

Hyperprior and Context model

(Ballé et al. 2018) posit that the elements of $\hat{y}$ exhibit a significant spatial scale dependence. To encapsulate this spatial dependence, they introduce an additional set of random variables, denoted as $\tilde{z}$ in Figure 2b. $\hat{y}$ is modeled as a Gaussian distribution with zero-mean and a standard deviation of σ. This standard deviation is predicted by applying the parametric transformation h_s to $\tilde{z}$. The compressed hyperprior could be added as edge information to the bitstream in persistent storage, which allows the synthesizer to use the conditional entropy model.

(Minnen, Ballé, and Toderici 2018), based on the work of (Ballé et al. 2018), joint the mean-scale hyperprior and added an autoregressive context module, denoted as C_m in Figure 2c. Although the autoregressive context model achieves better RD performance by correlating the currently decoded symbols with the already decoded symbols, its inability to perform parallel computation greatly affects the en/decode speed. Its usual coding and decoding time for a image reaches an intolerable number of seconds. Therefore, solving the parallelism problem of the context model becomes the key to improve the efficiency.

Parallel context model

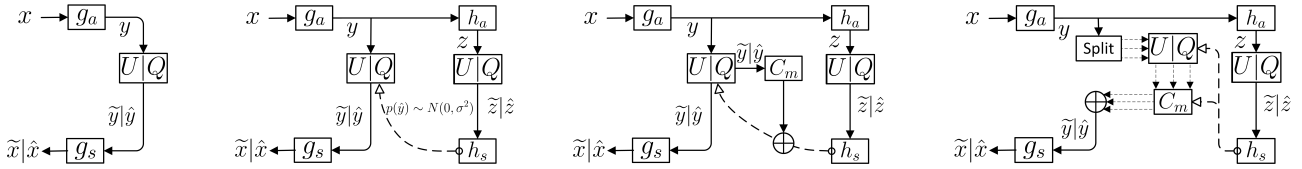
(He et al. 2021) proposes to separate symbols ($\hat{y} = \{\hat{y}_1, ..., \hat{y}_{h\times w}\}$) as anchors and non-anchors (Eq 2) and implements parallel decoding of anchors and non-anchors using a checkerboard space context convolution $\Phi_{sp,i} = g_{sp}(\hat{y}_{<i})$, $\hat{y}_{<i} = \{\hat{y}_1, ..., \hat{y}_{i-1}\}$.

$$\hat{y}_{<i}^{(anchor)} = \varnothing, \hat{y}_{<i}^{(nonanc)} = \hat{y}^{(anchor)} \quad (2)$$

(Minnen and Singh 2020) proposed a channel-wise context model. In this model, symbols ($\hat{y}$) are unevenly split into K chunks, as Figure 2d, and then all chunks are processed sequentially through autoregressive context decoding. Except for the first chunk, subsequent context decoding utilizes information from the already decoded channel chunks:

$$\Phi_{ch}^{(k)} = g_{ch}^{(k)}(\hat{y}^{<k}) \quad (3)$$

where, $\hat{y}^{<k} = \{\hat{y}^{(1)}, ..., \hat{y}^{(k-1)}\}$ represents the chunks that have already been decoded. They set $K = \{2, 3, 4, 5, 8, 10\}$ and confirmed through experimentation that larger K exhibit better RD performance, but with a corresponding decrease in decoding speed.



(a) Base (b) Hyperprior (c) Joint (d) Channel context

Figure 2: Learning image compression frameworks.

(He et al. 2022) proposed an unevenly grouped channel context model. They found that in the channel-wise context model (Minnen and Singh 2020), which is split into 10 groups, the feature maps with higher information entropy are distributed in the top channels, so they grouped them unevenly by $K = 5$, $G = \{16, 16, 32, 64, 192\}$. Each group is decoded separately using checkerboard context. Based on this, they propose a thumbnail synthesizer that decodes the first 128 channels to generate preview images.

Methods

The information entropy and homomorphism of the intermediate feature maps of neural network have an significant impact on the performance of image entropy coding. We find that the information entropy is similar for one-channel and multi-channel images after coding. However, the homomorphism of latent features is reduced due to inter-channel interactions in multi-channel images compared to one-channel images. We conjecture that multi-channel image compression mode is unfavorable for image entropy coding. Therefore, we propose a general format learned image compression model and accompanying modules.

One-channel learned image compression

Existing learned image compression techniques are modeling end-to-end rate distortion on three-channel input images. We try to transform an arbitrary channel image into a one-channel image by performing a tiling operation, and then perform entropy modeling. For instance, for a three-channel RGB image, we vertically stitched the input image $(3, H, W)$ into a one-channel image of $(1, 3 \times H, W)$ and fed it to the encoder g_a.

Generally, when the $U|Q$ module quantizes and entropy encodes y, the feature map size $(3 \times h, w)$ obtained from the one-channel model via g_a is much larger than the feature map size (h, w) of the three-channel model. We further calculate the Image Information Entropy (Eq 4) and Homogeneity Index (Eq 5) of y for both schemes.

$$\mathbb{H}(X) = -\sum_{i=0} p(x_i) \log_2 p(x_i) \tag{4}$$

where $p(x_i)$ denotes the probability of the i-th gray level in the image, obtained from the image histogram. Image information entropy , quantifies the complexity of an image based on its gray level or color distribution. High entropy images are rich in details and textures, whereas low entropy ones tend to be more uniform.

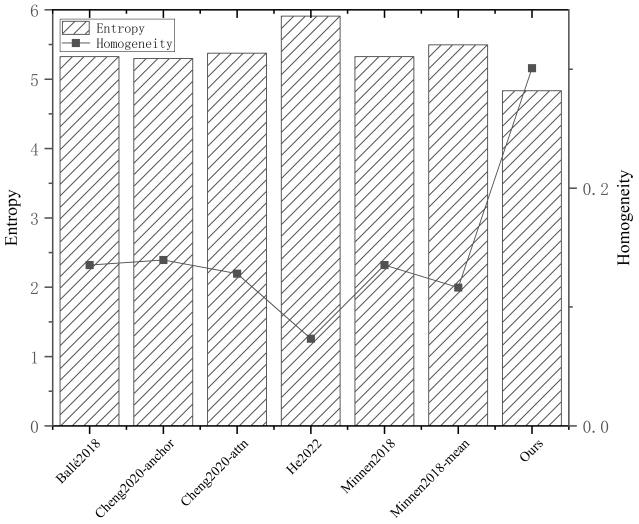


Figure 3: Information entropy and homogeneity. The results come from existing learned image compression models: Ballé2018 (Ballé et al. 2018), Cheng2020 (Cheng et al. 2020), He2022 (He et al. 2022), Minnen2018 (Minnen, Ballé, and Toderici 2018) and our proposed GLIC model. The results are run in *kodim11* (Eastman Kodak Company 1999) and the similar experimental results are obtained in other images.

$$f_{homogeneity} = \sum_{i,j} \frac{P(i,j)}{1 + (i-j)^2} \tag{5}$$

where $P(i, j)$ is the value of the Gray Level Co-occurrence Matrix (Haralick, Shanmugam, and Dinstein 1973) for the i-th row and j-th column, which represents the probability that a pixel with gray level i and a neighboring pixel with gray level j co-occur in the image. The homogeneity index is a metric used to measure the local consistency within an image. A higher homogeneity index indicates a more uniform texture.

Entropy coding compresses image by assigning shorter codes to frequently occurring symbols. Image information entropy provides a theoretical lower bound for information storage. Image entropy coding attempts to approach this lower bound through coding techniques to achieve efficient image compression.

Our experiments, as Figure 3, indicate that the one-channel model maintains the theoretical efficiency of en-

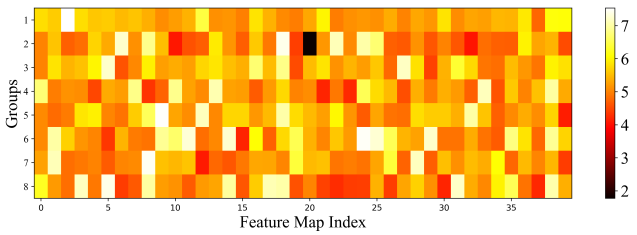


Figure 4: Information entropy distribution. This result is the Information Entropy Distribution of the $M = 320$ output feature maps obtained by g_a in *kodim11*. We obtained similar results for other images.

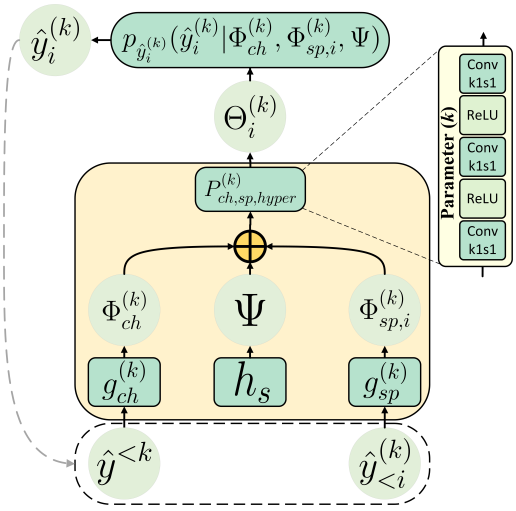


Figure 5: Cross-channel context.

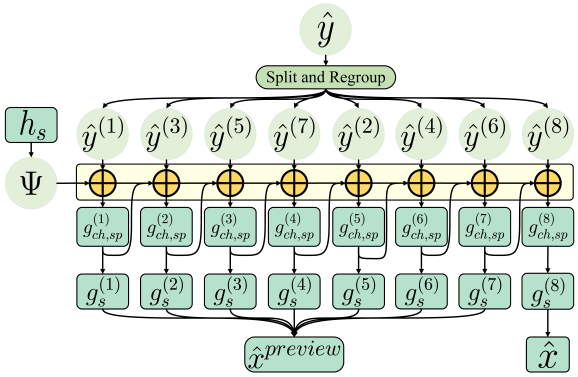


Figure 6: Progressive image decoding.

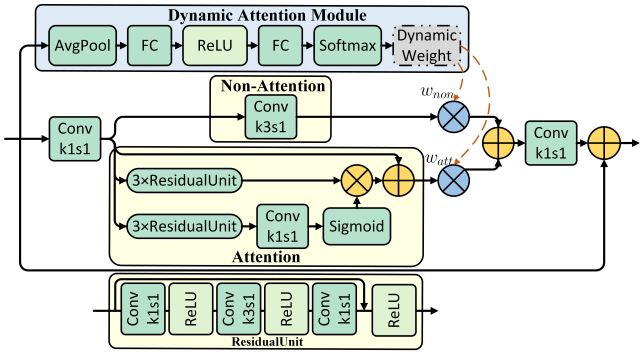


Figure 7: Adaptive attention residual block.

tropy coding, even with enlarged feature maps. However, analyzing the average homogeneity index of the output feature maps of g_a, the one-channel model preserves the distribution of approximate pixels (i.e., less noise, more homogeneous texture, and more continuous gray scale variation) more efficiently. This is attributed to multi-channel inputs altering the original pixel distribution during convolution. In contrast, a dense distribution of identical pixels is more favorable for image entropy coding. Thus, the approach aligns better with entropy coding principles, ensuring closeness to its theoretical efficiency.

Cross-channel context module

Module design We visualize the information entropy distribution of the output feature maps of the one-channel synthesizer g_a, as Figure 4. After splitting all the feature maps into 8 groups in index order, we statistic all the information entropy and find that the mean value of each group is quite close to each other. (He et al. 2022) confirmed that the feature map information entropy of the multi-channel image compression model exhibits an uneven distribution, and proposed to decode the first 4 chunks (128 channels) for progressive preview image decoding. Obviously, the one-channel learned image compression model is more suitable for the evenly divided channel method.

Based this analysis, we propose an evenly grouped cross-channel context (CCCT) Module, as Figure 5. The method is able to quickly obtain the global information of the image and then decode the complete information step by step. Specifically, evenly split the $M = 320$ channels of $\hat{y}$ into $K = 8$ chunks with 40 channels each, then sort the odd-indexed chunks in front of the even-indexed ones: $\hat{y}^{(Split)} = \{\hat{y}^{(1)}, \hat{y}^{(3)}, \hat{y}^{(5)}, \hat{y}^{(7)}, \hat{y}^{(2)}, \hat{y}^{(4)}, \hat{y}^{(6)}, \hat{y}^{(8)}\}$, $\hat{y}^{<k} = \{\hat{y}^{(1)}, \hat{y}^{(3)}, \hat{y}^{(5)}, ..., \hat{y}^{(k-1)}\}$. Then, we adopt (He et al. 2022) similar combining methodology, using checkerboard spatial context $g_{sp}^{(k)}$ (He et al. 2021) to obtain spatial redundancy $\Phi_{sp,i}^{(k)}$ for the k-th chunk, and the channel context $g_{ch}^{(k)}$ is used to obtain the channel redundancy $\Phi_{ch}^{(k)}$ for the channel chunks that have been decoded $\hat{y}^{<k}$. Further, the hyperprior parameter fusion module $P_{ch,sp,hyper}^{(k)}$ is used for $\Phi_{ch}^{(k)}$, $\Phi_{sp,i}^{(k)}$ and hyperprior information Ψ, in order to obtain the gaussian conditioning parameter $\Theta_i^{(k)} = (\mu_i^{(k)}, \sigma_i^{2(k)})$.

Finally, uniform noise $U(-\frac{1}{2}, \frac{1}{2}))$ is added to $\hat{y}^k$ (an alternative quantization operation (Ballé et al. 2018)) and entropy

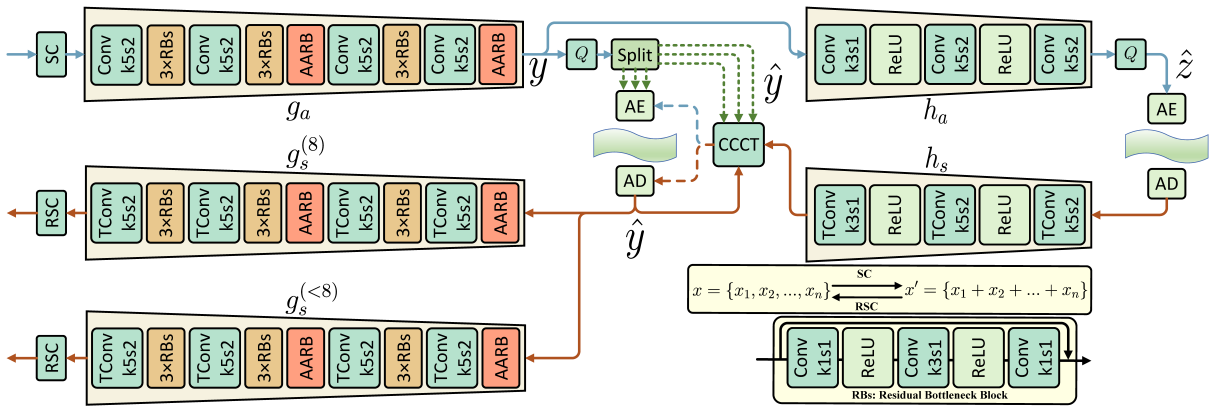


Figure 8: Network architecture. The Reconstruction Split and Concat (SC/RSC) module is used to split and reorganize images of any format into one-channel matrices (and inverse operations). The Q is a quantization operation (the training process is replaced by the addition of uniform noise). AE/AD is an arithmetic en/decode operation.

coding is performed using $\Theta_i^{(k)}$:

$$p_{\hat{y}_i^{(k)}|\Theta_i^{(k)}}(\hat{y}_i^{(k)}|\Phi_{ch}^{(k)}, \Phi_{sp,i}^{(k)}, \Psi) = (N(\mu_i^{(k)}, \sigma_i^{2(k)}) * U(-\frac{1}{2}, \frac{1}{2}))(\hat{y}_i^{(k)}) \quad (6)$$

Progressive image preview High-quality and fast progressive decoding is an important application of image compression techniques. In applications such as file previews, file thumbnails, and web image thumbnails, where only a low quality preview image needs to be displayed, there is a high demand for image decoding speed.

Based on the cross-channel context model, we propose a progressive image decoding technique: construct 8 levels of image synthesizer $g_s^{(1,2,3,4,5,6,7,8)}$, where g_s^k is used to decode $\hat{y}^{<(k+1)}$, i.e., to decode the information of $k \times 40$ channels. Specially, $g_s^{(8)}$ is performed for decoding the complete image information. When $g_s^{(8)}$ decodes the image, experiments confirm that the CCCT and the channel-wise context module achieve the same performance. However, further experiments confirm that the CCCT has better progressive preview ($g_s^{(<8)}$) image decoding performance than the channel-wise context module. As Figure 6, the potential representation $\hat{y}$ is divided into 8 groups, and when the k-th level of preview decoding is performed, $g_{ch}^{\{k\}}$ and $g_{sp}^{\{k\}}$ are used first to obtain the channel and spatial context information for the k chunks. Then preview decoding is performed using $g_s^{\{k\}}$ to get the preview image.

Our experiments also confirm that prioritizing the decoding of even-channel blocks before odd-channel blocks achieves the same model performance. This paper focuses on a more intuitive implementation.

Adaptive Attention Residual Block

(Cheng et al. 2020) proposed that the Residual Attention Block (RAB) can effectively improve the RD performance. This module has significant effect on high-level semantic information, however, when multiple Residual Bottleneck Block (RBs) stacks model the overall probability distribution of an image, the RAB needs to extract the high-level semantic information while keeping the low-level semantic information intact, regardless of the importance of these semantic information.

Inspired by the image super-resolution net AAN (Chen, Gu, and Zhang 2021), we combine the Attention in Attention Block (AAB) (Chen, Gu, and Zhang 2021) and the RAB (Cheng et al. 2020) to propose an Adaptive Attention Residual Block (AARB) module that adaptively distinguishes the importance of semantic information, as Figure 7. The AARB module separates the Attention module and lets it focus on extracting high-level semantic information. The Non-Attention module is utilized to retain the overall semantic information. At last, the importance of the two kinds of semantic information is decided adaptively. Dynamic Attention Module is used to adaptively assign the importance of different levels of semantic information. Softmax module outputs two weight parameters, $w_{non} + w_{att} = 1$.

Analyzer g_a	Synthesizer $g_s^{(k)}$
input:1 channel	input:$k \times 40$ channel
output:320 channel	output:1 channel
Conv 5×5, s2, N	AARB, $k \times 40$
ResBottleneck $\times$ 3	TConv 5×5, s2, $k \times 24$
Conv 5×5, s2, N	ResBottleneck $\times$ 3
ResBottleneck $\times$ 3	TConv 5×5, s2, $k \times 24$
AARB, N	AARB, $k \times 24$
Conv 5×5, s2, N	ResBottleneck $\times$ 3
ResBottleneck $\times$ 3	TConv 5×5, s2, $k \times 24$
Conv 5×5, s2, M	ResBottleneck $\times$ 3
AARB, M	TConv 5×5, s2, 1

Table 1: Network details. $k \in \{1, 2, 3, 4, 5, 6, 7, 8\}$

Hyperprior analyzer h_a	Hyperprior synthesizer h_s
input:320 channel	input:192 channel
output:192 channel	output:640 channel
Conv 3 × 3, s1, N	TConv 5 × 5, s2, N
ReLu	ReLu
Conv 5 × 5, s2, N	TConv 5 × 5, s2, N × 1.5
ReLu	ReLu
Conv 5 × 5, s2, N	TConv 3 × 3, s1, M × 2

Table 2: Hyperprior network details.

Architecture

Our proposed methods are all integrated in the GLIC model as Figure 8. Through the SC/RSC module with an improved network channel connection structure, our method is able to adapt to image compression tasks in any format. The CCCT module successfully achieves efficient parallel computation by decoding the reorganized symbols $\hat{y}$ step by step. Based on the priority of cross-channel decoding to obtain the overall semantic information, according to the practical application requirements, the progressive image decoding can be achieved using the preview image synthesiser $g_s^{(<8)}$, which is extremely fast in preview image decoding, however, its decoding quality is only slightly lower than the complete decoding. The difference between $g_s^{(<8)}$ compared to $g_s^{(8)}$ is only in the width of the network, the size of the output preview image is the same as the size of the original image.

Experiments

Comprehensive experiments are conducted in order to demonstrate the superiority of GLIC in the following aspects: the RD performance, the effectiveness of arbitrary format image compression applications, the reliability of progressive decoding, and the validity of subjective visual judgment. In addition, to verify the effectiveness of the proposed one-channel scheme, the AARB module, and the CCCT module, we perform ablation studies.

Code — https://github.com/DuBianJun-007/GLIC-General-Format-Learned-Image-Compression

Model design details

Figure 8 gives a graphical representation of the GLIC network architecture. Table 1 and Table 2 give the detailed configuration of the network. We use $N = 192$ and $M = 320$ to denote the number of intermediate channels in the network. The input and output channels become one-channel tensor $(1 \times H \times W)$.

Figure 8 gives the Reconstruction Split and Concat (SC/RSC) module, which is an image preprocessing operation in which the SC module tiles an arbitrary channel image (C, H, W) to obtain a one-channel image matrix, which can have the shape $(1, C \times H, W)$ or $(1, H, C \times W)$ or $(1, C^1 \times H, C^2 \times W)$, where $C^1 \times C^2 = C$. The SC module needs to save the shape parameters of the original image after the image preprocessing is completed for the RSC module to reorganize the decoded image to be restored to the original image format.

Our code examples uploaded on GitHub give the above three preprocessing implementation schemes. In practical application scenarios, arbitrary reducible image flattening operations can be performed according to different application requirements.

Implementation details

During model training, we use a subset of the ImageNet dataset (Deng et al. 2009), including 14,206 images for training and 3,465 images for testing. These images require preprocessing operations to unify the format into a one-channel image as training input. We use the Mean Square Error (MSE) as the distortion term for optimization. The optimizer we employ is Adam (Kingma and Ba 2014). The hyperparameter λ is set to these values: $\{0.002, 0.0042, 0.0075, 0.015, 0.03, 0.045, 0.07, 0.09\}$. Using a learning rate of 1×10^{-4} and *batch_size*$=16$, the main network is trained for 100 epochs. Subsequently, 2000 epochs of fine-tuning with a learning rate of 1×10^{-5} ensure its convergence.

After the above training, all parameters of the main network are frozen, only the gradient of $g_s^{(<8)}$ is turned on, and independent training is performed using a learning rate of 1×10^{-5}. Each preview synthesizer requires 2000 epochs,the lower the preview level,the more training time is required.

Data set	Channel	BPP	PSNR	MS-SSIM (dB)
LUNA16	Multi-C	1.05	41.43	25.58
MURA	1	0.41	44.79	25.91
Bluetooth	1	3.34	37.75	22.79

Table 3: Multi-channel image compression. The number of image channels in the LUNA16 dataset varies from tens to hundreds. The MURA dataset consists of one-channel images. The Bluetooth dataset is derived from spectral I/Q data and is also one-channel.

Rate-distortion performance

As Figure 9, we validated the BPP-PSNR and BPP-MS-SSIM (dB) of the GLIC model in the Kodak (Eastman Kodak Company 1999), with MS-SSIM converted to decibel representation $(-10\log_{10}(1 - \text{MS-SSIM}))$. We benchmark GLIC with popular learned image compression models and hand-crafted image compression algorithms. A portion of the experimental data came from CompressAI (Bégaint et al. 2020). Experimental results indicate that GLIC outperforms existing models in terms of RD performance. This further affirms that the one-channel model preserves the homomorphism of image information more efficiently than the multi-channel model. Additionally, the information entropy of the intermediate feature maps remains unchanged after flattening the image into a large one-channel image.

Preview synthesizer

Each model derived from different λ has 8 synthesizers $g_s^{\{1,2,3,4,5,6,7,8\}}$. These synthesizers are responsible for de-

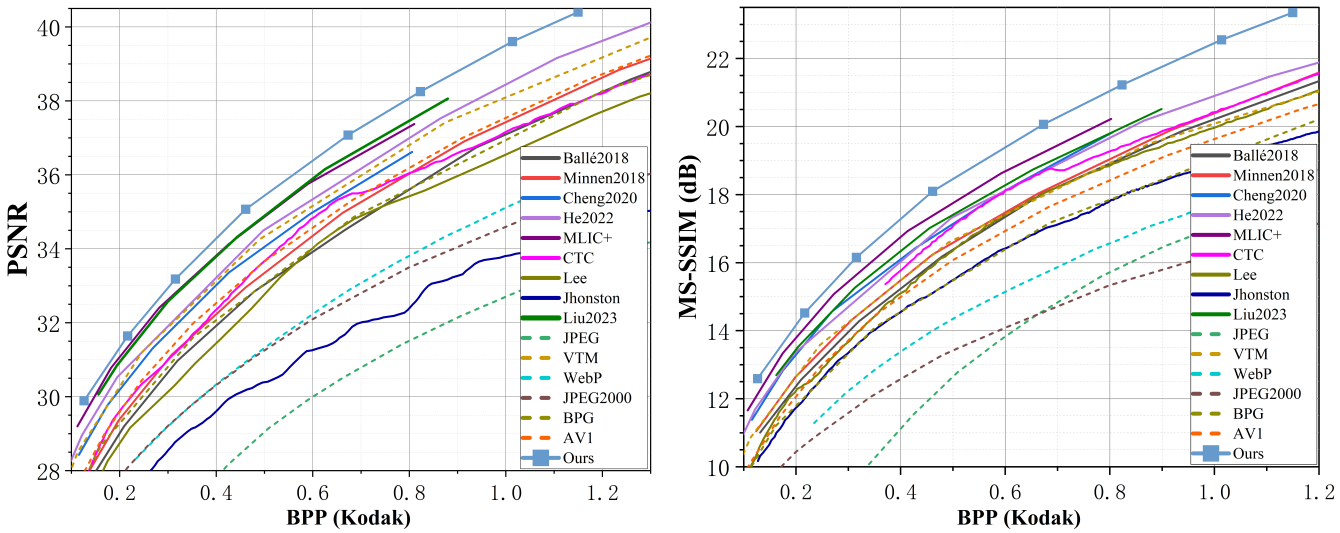


Figure 9: Rate distortion performance. The results are derived from various image compression methods: Ballé2018 (Ballé et al. 2018), Minnen2018 (Minnen, Ballé, and Toderici 2018), Cheng2020 (Cheng et al. 2020), He2022 (He et al. 2022), MLIC+ (Jiang et al. 2023), CTC (Jeon et al. 2023), Lee (Lee et al. 2022), Jhonston (Johnston et al. 2018), Liu2023 (Liu, Sun, and Katto 2023), JPEG (Pennebaker and Mitchell 1992), VTM (Wien and Bross 2020), WebP (Pintus et al. 2012), JPEG2000 (Taubman and Marcellin 2002), BPG (Bellard 2015), AV1 (Norkin et al. 2022) and our proposed GLIC model.

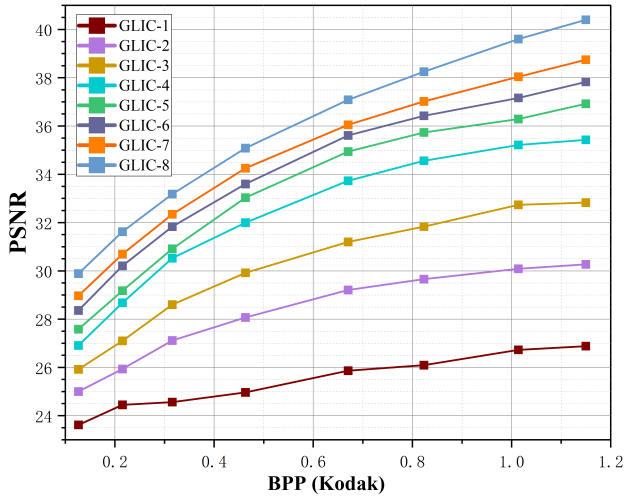


Figure 10: Progressive image decoding on Kodak.

coding various numbers of chunks. As Figure 10, different synthesizers demonstrate the progressive decoding RD performance. The advantage of lower quality decoding is that it provides a quick preview of the image. Notably, $g_s^{\{1,2,3\}}$ did not capture the overall features of the images, resulting in a significant RD performance gap with the higher-quality synthesizer $g_s^{\{>3\}}$. Conversely, $g_s^{\{>3\}}$ shows a consistent trend of improved performance.

General format results

The general format learned image compression technique is a downscaling solution designed for image compression across various channels, not just limited to three-channel images. As Table 3, GLIC has seen successful implementation in multiple image formats with $\lambda = 0.09$ for the model.

For example, this method has undergone testing on specialized medical imaging datasets, specifically the *subset0* dataset from Lung Nodule Analysis 2016 (LUNA16) (Setio et al. 2017), and the *validation set* from the Musculoskeletal Radiographs (MURA) (Rajpurkar et al. 2018). Beyond the realm of medical imaging, GLIC can also be extended to archive large volumes of historical Radio Frequency (RF) data in electromagnetic spectrum management. An example of this involves the In-phase and Quadrature phase (I/Q) data from Bluetooth devices (Uzundurukan, Dalveren, and Kara 2020). In this scenario, we convert the I/Q data into a one-channel matrix (grayscale image) for storage. These outcomes verify the effectiveness of GLIC in handling images of arbitrary channel, demonstrating its potential as a general format image compression solution.

Qualitative results

Subjective visual evaluation is an important qualitative metric for judging the quality of lossy image compression. Therefore, we perform a visual comparison. The focus is on comparing the texture details preserved in decoded images at similar compression bit rates. As Figure 11, shows the reconstructed image and details. Compared to other methods, GLIC has better RD performance and visual discrimination, and our method preserves more texture details.

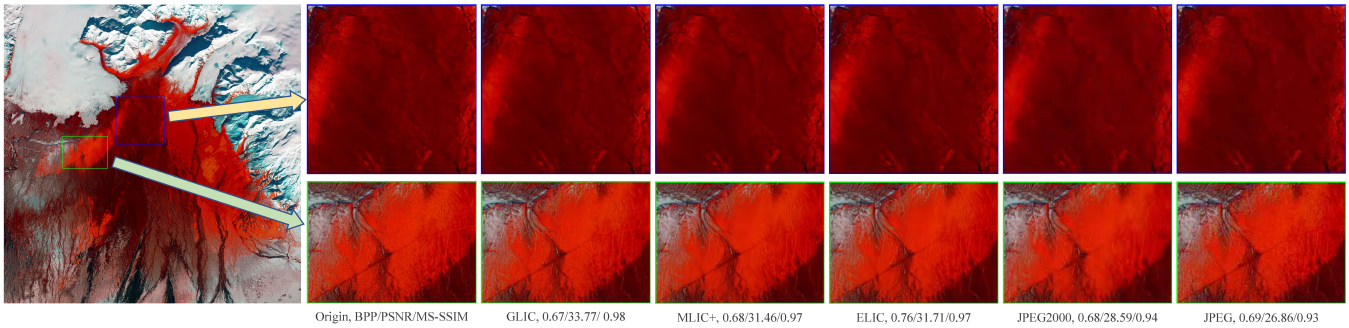
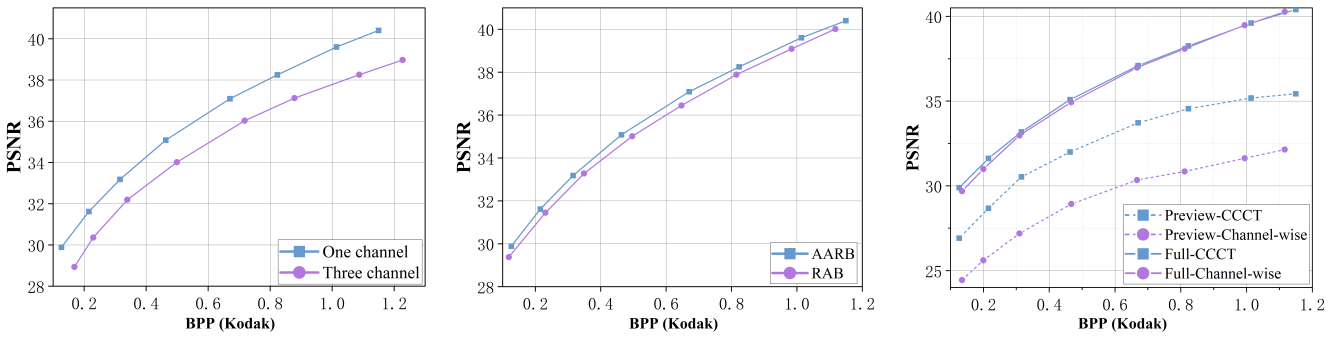


Figure 11: Comparison of reconstructions of *c26847af1470e880236db4766b42c09d.png* (CLIC Organizing Committee 2024).



(a) Channel number. (b) AARB and RAB. (c) Context module.

Figure 12: Ablation experiment.

Ablation Study

Number of channels To confirm the one-channel model's superiority for entropy coding, we compared it with the traditional three-channel model. We keep the structure of the main, hyperpriori, and context modules constant to control for ablation variables, the only difference being the number of model channels. As Figure 12a gives the evaluation results on Kodak(Eastman Kodak Company 1999). The RD performance of the one-channel model is significantly higher than that of the three-channel model. This strengthens the assertion that the multi-channel model's homomorphism feature is negatively impacted by inter-channel information.

Furthermore, we observe an unusual phenomenon: when the multi-channel model is paired with the CCCT, the model inconsistently fails to encode and decode the same image, meaning it can't reliably compress images. However, when the multi-channel model is used alongside the channel-wise module, this issue doesn't occur. This is the reason why the RD performance plummets when the three-channel model is combined with CCCT in the experimental results. We consider that the multi-channel model produces significant channel correlation, which can be eliminated by CCCT.

To summarize, it is clear that channel interactions can negatively affect entropy coding. The one-channel model solves this problem by eliminating this strong channel correlation.

AARB and RAB module To verify the effectiveness of our improve AARB module in combination with the one-channel model. We replace only the AARB with the RAB proposed by (Cheng et al. 2020) to train the model with 8 levels, where the hyperparameters λ are set to: $\{0.0015, 0.004, 0.008, 0.015, 0.025, 0.04, 0.06, 0.08\}$. The RD curves are shown in Figure 12b. The AARB module combined with the one-channel model structure achieves better RD performance. Combined with the model structure analysis, it can be seen that the attention branch of AARB has the same ability to extract local deep semantic information as the RAB module, and the non-attention branch can better retain the overall semantic information. The adaptive module weight assignment block can effectively discriminate the importance of overall and local semantic information.

CCCT and Channel-wise context module The difference between CCCT and channel-wise context module (Minnen and Singh 2020) is that CCCT can earlier extract the overall information of the image and then gradually obtain the complete information. To verify the effectiveness of CCCT, we only replace the context module for experiments to compare the RD performance of the preview image for decoding half of the chunks, i.e., $g_s^{(4)}$. As Figure 12c gives the conclusions drawn on Kodak (Eastman Kodak Company 1999). The complete decoded image RD performance of both context modules is almost the same, but the preview image RD performance of CCCT is better than channel-wise context. This result confirms that CCCT is able to capture the overall image information earlier and is more suitable for progressive image decoding tasks.

Conclusion

We propose the General Format Learned Image Compression (GLIC) model, which can be widely applied to various image compression tasks. We analyze the intermediate feature maps of the multi-channel and the one-channel model, which reveals that separating the inter-channel correlations can enhance the homogeneity of the feature maps without reducing their information entropy. This is more favorable for image entropy coding. Therefore, we propose a scheme to unify any images into a general format, and combine it with an improved Adaptive Attention Residual Block to achieve excellent rate-distortion performance. Further, we propose Cross-Channel Context Module, which quickly captures the overall semantic information and realizes high-quality progressive preview image decoding. Our experimental results show that GLIC achieves excellent improvements in both rate-distortion performance and model applicability.

Acknowledgments

This work was funded by Intelligent Policing Key Laboratory of Sichuan Province (ZNJW2024KFZD004), the key R&D project jointly implemented by Sichuan and Chongqing in 2020 (cstc2020jscx-cylhX0004), and the Intelligent Policing and National Security Risk Management Laboratory (ZHZZZD2301).

References

Ali, M. S.; Kim, Y.; Qamar, M.; Lim, S.-C.; Kim, D.; Zhang, C.; Bae, S.-H.; and Kim, H. Y. 2024. Towards efficient image compression without autoregressive models. *Advances in Neural Information Processing Systems*, 36.

Ballé, J.; Chou, P. A.; Minnen, D.; Singh, S.; Johnston, N.; Agustsson, E.; Hwang, S. J.; and Toderici, G. 2020. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2): 339–353.

Ballé, J.; Laparra, V.; and Simoncelli, E. P. 2016. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*.

Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*.

Bégaint, J.; Racapé, F.; Feltman, S.; and Pushparaja, A. 2020. CompressAI: a PyTorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*.

Bellard, F. 2015. BPG image format. *URL https://bellard.org/bpg*, 1(2): 1.

Chen, H.; Gu, J.; and Zhang, Z. 2021. Attention in attention network for image super-resolution. *arXiv preprint arXiv:2104.09497*.

Cheng, Z.; Sun, H.; Takeuchi, M.; and Katto, J. 2020. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7939–7948.

CLIC Organizing Committee. 2024. Workshop and Challenge on Learned Image Compression. [Online]. Available: http://compression.cc/tasks/. The 6th Challenge on Learned Image Compression.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Eastman Kodak Company. 1999. Kodak Lossless True Color Image Suite. Download from http://r0k.us/graphics/kodak/. Accessed: November 15, 1999.

Gao, C.; Xu, T.; He, D.; Wang, Y.; and Qin, H. 2022. Flexible neural image compression via code editing. *Advances in Neural Information Processing Systems*, 35: 12184–12196.

Gonzales, R. C.; and Wintz, P. 1987. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc.

Goyal, V. K. 2001. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, 18(5): 9–21.

Haralick, R. M.; Shanmugam, K.; and Dinstein, I. H. 1973. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6): 610–621.

He, D.; Yang, Z.; Peng, W.; Ma, R.; Qin, H.; and Wang, Y. 2022. 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*, 5718–5727.

He, D.; Zheng, Y.; Sun, B.; Wang, Y.; and Qin, H. 2021. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14771–14780.

Jeon, S.; Choi, K. P.; Park, Y.; and Kim, C.-S. 2023. Context-Based Trit-Plane Coding for Progressive Image Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14348–14357.

Jiang, W.; Yang, J.; Zhai, Y.; Ning, P.; Gao, F.; and Wang, R. 2023. Mlic: Multi-reference entropy model for learned image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, 7618–7627.

Johnston, N.; Vincent, D.; Minnen, D.; Covell, M.; Singh, S.; Chinen, T.; Hwang, S. J.; Shor, J.; and Toderici, G. 2018. 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*, 4385–4393.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, J.; Jeong, S.; and Kim, M. 2022. Selective compression learning of latent representations for variable-rate image compression. *Advances in Neural Information Processing Systems*, 35: 13146–13157.

Lee, J.-H.; Jeon, S.; Choi, K. P.; Park, Y.; and Kim, C.-S. 2022. DPICT: Deep progressive image compression using trit-planes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16113–16122.

Li, Y.; Xu, T.; Wang, Y.; Liu, J.; and Zhang, Y.-Q. 2024. Idempotent Learned Image Compression with Right-Inverse. *Advances in Neural Information Processing Systems*, 36.

Lieberman, K.; Diffenderfer, J.; Godfrey, C.; and Kailkhura, B. 2023. Neural Image Compression: Generalization, Robustness, and Spectral Biases. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*.

Liu, J.; Sun, H.; and Katto, J. 2023. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14388–14397.

Martin, G. N. N. 1979. 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*, volume 2.

Mentzer, F.; Agustsson, E.; Tschannen, M.; Timofte, R.; and Van Gool, L. 2018. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4394–4402.

Minnen, D.; Ballé, J.; and Toderici, G. D. 2018. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31.

Minnen, D.; and Singh, S. 2020. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, 3339–3343. IEEE.

Nian, Y.; Liu, Y.; and Ye, Z. 2016. Pairwise KLT-based compression for multispectral images. *Sensing and Imaging*, 17: 1–15.

Norkin, A.; Grange, A.; Concolato, C.; Katsavounidis, I.; Tmar, H.; Mammou, K.; Liu, S.; and Baliga, R. 2022. Alliance for open media (aomedia) progress report. *SMPTE Motion Imaging Journal*, 131(8): 88–92.

Pennebaker, W. B.; and Mitchell, J. L. 1992. *JPEG: Still image data compression standard*. Springer Science & Business Media.

Pintus, M.; Ginesu, G.; Atzori, L.; and Giusto, D. D. 2012. Objective evaluation of webp image compression efficiency. In *Mobile Multimedia Communications: 7th International ICST Conference, MOBIMEDIA 2011, Cagliari, Italy, September 5-7, 2011, Revised Selected Papers 7*, 252–265. Springer.

Rajpurkar, P.; Irvin, J.; Bagul, A.; Ding, D.; Duan, T.; Mehta, H.; Yang, B.; Zhu, K.; Laird, D.; Ball, R. L.; Langlotz, C.; Shpanskaya, K.; Lungren, M. P.; and Ng, A. Y. 2018. MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs. arXiv:1712.06957.

Rezasoltani, S.; and Qureshi, F. Z. 2023. Hyperspectral Image Compression Using Implicit Neural Representations. In *2023 20th Conference on Robots and Vision (CRV)*, 248–255. IEEE.

Rissanen, J.; and Langdon, G. 1981. Universal modeling and coding. *IEEE Transactions on Information Theory*, 27(1): 12–23.

Setio, A. A. A.; Traverso, A.; De Bel, T.; Berens, M. S.; Van Den Bogaard, C.; Cerello, P.; Chen, H.; Dou, Q.; Fantacci, M. E.; Geurts, B.; et al. 2017. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge. *Medical image analysis*, 42: 1–13.

Taubman, D. S.; and Marcellin, M. W. 2002. JPEG2000: Standard for interactive imaging. *Proceedings of the IEEE*, 90(8): 1336–1357.

Uzundurukan, E.; Dalveren, Y.; and Kara, A. 2020. A Database for the Radio Frequency Fingerprinting of Bluetooth Devices. *Data*, 5(2).

Van Leeuwen, J. 1976. On the Construction of Huffman Trees. In *ICALP*, 382–410.

Wang, Z.; Simoncelli, E. P.; and Bovik, A. C. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, 1398–1402. Ieee.

Wien, M.; and Bross, B. 2020. Versatile video coding–algorithms and specification. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 1–3. IEEE.

Xu, T.; Wang, Y.; He, D.; Gao, C.; Gao, H.; Liu, K.; and Qin, H. 2022. Multi-sample training for neural image compression. *Advances in Neural Information Processing Systems*, 35: 1502–1515.

Yang, R.; and Mandt, S. 2024. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36.