Learned Image Compression via Sparse Attention and Adaptive Frequency

Huidong Ma[1,2], Xinyan Shi[1], Hui Sun[1*], Xiaofei Yue[3], Xiaoguang Liu[1*], Gang Wang[1*], Wentong Cai[2]
[1] College of Computer Science, TMCC, SysNet, DISSec, GTIISC, Nankai University, Tianjin, China
[2] College of Computing and Data Science, Nanyang Technological University, Singapore
[3] Beijing Institute of Technology, Beijing, China
{mahd, sunh, liuxg, wgzwp}@nbjl.nankai.edu.cn

Abstract

Learned image compression (LIC) methods surpass traditional algorithms in rate-distortion (RD) performance, but still struggle to optimally balance effectiveness and efficiency. Moreover, although recent studies have demonstrated the effectiveness of utilizing frequency-domain information, they typically rely on fixed frequency transforms and still lack content-adaptive capabilities. Therefore, we propose an advanced spatial-frequency dual-path LIC method with enhanced adaptivity and efficiency. Specifically, for the spatial path, we introduce Cross-Sparse Window Attention, leveraging sparse, window-conditioned global tokens to efficiently model long-range dependencies. It achieves lower computational cost and superior effectiveness than standard Window-based Multi-head Self-attention. For the frequency path, we design a content-adaptive frequency transform, employing a decomposition weight generator and learnable global weights to adaptively process multi-scale frequency components. Furthermore, we propose Denoising-as-Regularizer, a training-only module that structures and smooths the latent representation via a denoising task, enhancing reconstruction quality at zero inference cost. Experiments on the Kodak, CLIC, and Tecnick datasets demonstrate that the proposed method significantly outperforms existing state-of-the-art methods in both RD performance and latency.

1. Introduction

The rapid and explosive growth of digital image data across diverse fields, including social media, medical imaging, and AI-generated content, poses enormous storage and transmission challenges for modern network infrastructures. Image compression, which aims to represent image information using the fewest possible bits, is a key solution to address these challenges [31, 32, 42]. For decades, image compression was dominated by traditional standards like JPEG [45] and VVC [8], which use fixed modules for Transform, Quantization, and Entropy Coding. These methods suffer from fixed transforms unsuited for diverse natural images and independently optimized components, leading to artifacts at low bitrates. Recently emerging learned image compression (LIC) methods have addressed these issues. Toderici et al. [43] first showed an end-to-end optimized recurrent neural network (RNN) could achieve variable rates. Subsequently, Ballé et al. [5] established the standard autoencoder framework comprising two key components: a learned transform network and a learned entropy model, optimized jointly for rate-distortion (RD). Since then, algorithms have primarily focused on optimizing these two components. The transform network maps images to a compact latent space, determining the reconstruction quality (distortion). Some studies [9, 27, 38, 39, 47, 48, 50] have focused on enhancing the transform network's architecture, integrating potent modules like CNN-Transformers [44], invertible networks, and State Space Models [15] to learn more efficient and compact representations that minimize information loss. The entropy model is a learnable probabilistic framework that estimates the distribution of the quantized latents, thereby controlling the Rate component of the RD objective. Studies [6, 10, 35] have focused on enhancing predictive power, progressing from simple global priors to hyperprior and autoregressive models (to capture spatial dependencies), and ultimately to more sophisticated techniques using advanced contexts, attention, or Gaussian Mixture Models. Meanwhile, other studies [13, 17, 18, 25, 38, 47, 50] focus on improving training efficiency and inference speed for practical applications. These approaches, such as parallel-friendly context models, linear attention, frequency-aware modules, and decoupled training strategies, aim to reduce decoding latency and computational complexity.

*Corresponding authors. The source code is available at SAAF.

Despite significant progress in LIC, existing methods face several limitations that hinder the optimal trade-off between RD performance and inference speed. Standard Window-based Multi-head Self-attention (WMSA) used by some LIC methods [25, 27, 29] is confined to a local receptive field, while alternatives like shifted windows introduce higher computational complexity. Most LIC methods overlook multi-scale frequency patterns, which are crucial for image compression. Even the few recent studies [14, 24, 25] that incorporate frequency transforms rely on fixed transformations, which cannot achieve content-adaptation. To address these challenges, we propose the **S**parse **A**ttention and **A**daptive **F**requency for learned image compression (**SAAF**), a novel framework featuring three key components: 1) To resolve the spatial trade-off, we introduce Cross-Sparse Window Attention, a local-global mechanism featuring an innovative Global Sparse Attention (GSA). GSA uses a small set of window-conditioned learnable tokens to efficiently capture global dependencies at minimal cost. 2) To address the frequency limitation, we propose the Adaptive Frequency Block. It utilizes dynamic local weights and learnable global weights to dynamically decompose features into simulated frequency bands, enabling a fully content-adaptive frequency response. 3) To further improve RD performance, we introduce the Denoising-as-Regularizer module. Functioning as a training-only regularizer, it leverages diffusion principles to train a lightweight conditional noise predictor to estimate injected noise. This process forces the latent representation to become more structured and smooth, enhancing reconstruction quality at zero inference cost. Our contributions can be summarized into the following four points:

- We propose the Sparse Attention Block with novel Cross-Sparse Window Attention, a novel local-global attention mechanism that uses window-conditioned sparse global tokens to efficiently capture long-range dependencies at a lower computational cost.
- We propose the Adaptive Frequency Block, which introduces a learnable, content-adaptive transform to dynamically process multi-scale frequency components.
- We introduce Denoising-as-Regularizer, a training-only denoising-based strategy that regularizes the structure of the learned latent space to enhance RD performance with zero additional inference cost.
- Through extensive experiments, we demonstrate that our method outperforms existing advanced LIC methods in terms of both RD performance and speed.

2. Related Works

2.1. Transform Network

Current end-to-end architectures typically consist of a transform network and an entropy model. The transform networks in early LIC methods are predominantly based on Convolutional Neural Networks (CNNs). The pioneering work by Ballé et al. [5] first introduced CNN-based nonlinear transforms. He et al. [17] proposed a checkerboard-like context dependency design to improve decoding parallelism while maintaining performance. To overcome the limitations of CNNs in capturing long-range dependencies, some research has introduced attention mechanisms into LIC. Liu et al. [27] proposed a hybrid architecture of CNN and Transformer to extract local features and model global information. Zou et al. [52] introduced window-based attention to address the high computational complexity of Transformers on high-resolution images. DCAE [29] introduced a learnable dictionary into the Transformer-based network and provided priors during the entropy coding stage. GABIC [41] explores graph-based attention using k-nearest neighbors to optimize feature representation. More recently, the focus has shifted towards models with linear complexity for greater efficiency; this trend includes the adoption of State Space Models (SSMs) [15] in architectures like MambaVC [38], CMamba [47] and MambaIC [50], and the introduction of Bi-RWKV-based [36] models like LALIC [13], which demonstrate significant throughput and memory advantages over Transformers, particularly for processing high-resolution content.

2.2. Entropy Model

The entropy model is tasked with estimating the probability distribution of quantized latent representations to enable efficient compression. Early research evolved from simple factorized priors [5] to the landmark Hyperprior framework [6], which captures global statistics using side information and became a cornerstone for subsequent work. To further exploit local context, the hyperprior was combined with autoregressive models [35]; however, the inherent serial decoding bottleneck of this approach prompted the development of more parallel-friendly schemes, such as channel-wise autoregression [34] and checkerboard context models [17], etc. Beyond these foundational models, recent efforts have focused on more powerful context modeling, leveraging Transformers like in Entroformer [37] to capture long-range dependencies. A parallel line of research has focused on improving the probability model itself, replacing the single-Gaussian assumption with more expressive distributions like Gaussian Mixture Models (GMMs) [10, 51] to better fit the true latent distribution and directly improve rate-distortion performance.

2.3. Frequency Feature in LIC

A prominent research direction in learned image compression leverages frequency-domain analysis to treat an image's low- and high-frequency components differently. This approach evolved from early methods using explicit fre-

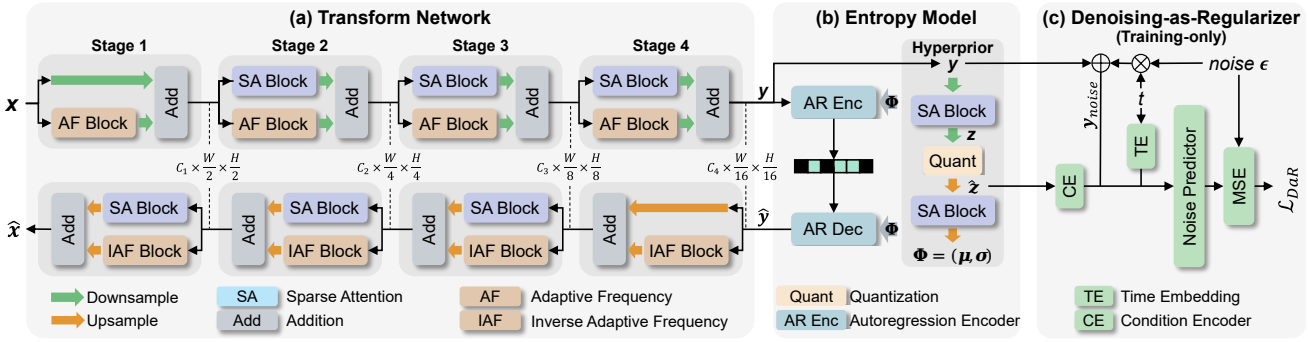

Figure 1. The workflow of the proposed LIC method SAAF. First, the transform network (encoder) hierarchically extracts features from the input image x to generate the latent representation $\boldsymbol{y}$. Subsequently, a hyperprior module extracts the prior information $\boldsymbol{\Phi} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$ from $\boldsymbol{y}$. Based on $\boldsymbol{\Phi}$, an autoregressive framework performs Gaussian modeling on $\boldsymbol{y}$ to guide the subsequent entropy coding (compression), while $\boldsymbol{y}$ is quantized into $\hat{\boldsymbol{y}}$. The inverse transform network (decoder) then reconstructs the image $\hat{\boldsymbol{x}}$ from $\hat{\boldsymbol{y}}$. Within the entropy model, our proposed DaR module learns a more structured latent space distribution by adding noise to the latent variable $\boldsymbol{y}$ and training the model to predict it. And it also enhances the model's compression quality and generalization ability.

quency decomposition, for instance by performing convolution directly in the wavelet domain [14, 33] or by designing learnable, end-to-end wavelet-like transforms [30]. A more deeply integrated strategy emerged with Generalized Octave Convolutions [1], a foundational paradigm that factorizes feature maps into multi-frequency groups within the network. This concept was subsequently extended to support multi-resolution learning and variable-rate coding [2, 26]. More recently, the broader principle of feature disentanglement has gained traction, inspiring methods that explicitly separate features by frequency [49] and others that apply disentangled training to the non-linear transform itself [25]. These ideas have culminated in advanced architectures like the Frequency-Aware Transformer [24], which combines frequency analysis with powerful attention mechanisms to optimize the rate-distortion trade-off.

3. Methods

3.1. Preliminaries

Modern learned image compression follows the hyperprior framework [6], which formulates compression as a rate-distortion optimization problem:

$$\mathcal{L}_{RD} = \mathbb{E}[R + \lambda \cdot D(\boldsymbol{x}, \hat{\boldsymbol{x}})] \tag{1}$$

where R denotes bitrate, D measures distortion (e.g., MSE), and λ controls the rate-distortion trade-off.

The architecture consists of a main encoder-decoder and a hyperprior module. The main transform uses encoder g_a and decoder g_s to compress the input image $\boldsymbol{x}$ into a latent representation $\boldsymbol{y}$, which is quantized to $\hat{\boldsymbol{y}}$ for entropy coding and then decoded to reconstruct $\hat{\boldsymbol{x}}$:

$$\boldsymbol{y} = g_a(\boldsymbol{x}), \quad \hat{\boldsymbol{y}} = Q(\boldsymbol{y}), \quad \hat{\boldsymbol{x}} = g_s(\hat{\boldsymbol{y}}) \tag{2}$$

To accurately model the spatial dependencies in $\hat{\boldsymbol{y}}$, a hyperprior autoencoder is introduced. The hyper-encoder h_a further compresses $\boldsymbol{y}$ into a hyper-latent $\boldsymbol{z}$ that captures local spatial statistics. After quantization to $\hat{\boldsymbol{z}}$, the hyper-decoder h_s estimates the mean $\boldsymbol{\mu}$ and scale $\boldsymbol{\sigma}$ for modeling each latent element $\hat{\boldsymbol{y}}_i$ with a Gaussian distribution:

$$\boldsymbol{z} = h_a(\boldsymbol{y}), \quad \hat{\boldsymbol{z}} = Q(\boldsymbol{z}), \quad (\boldsymbol{\mu}, \boldsymbol{\sigma}) = h_s(\hat{\boldsymbol{z}}) \tag{3}$$

$$P_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}_i|\hat{\boldsymbol{z}}) = \mathcal{N}(\hat{\boldsymbol{y}}_i; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2) \tag{4}$$

To further exploit spatial correlations, advanced methods introduce channel-wise autoregressive context [18, 35]. The latent $\hat{\boldsymbol{y}}$ is partitioned into spatial chunks (e.g., a checkerboard pattern), and within each chunk, channels are encoded sequentially. For the i-th channel, a context model aggregates information from previously decoded channels to refine the distribution parameters:

$$(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = \text{Context}(\hat{\boldsymbol{z}}, \hat{y}_{<i}) \tag{5}$$

where $\hat{y}_{<i}$ denotes already decoded channels. This autoregressive modeling significantly improves rate estimation by capturing inter-channel dependencies.

The hyper-latent $\hat{\boldsymbol{z}}$ itself is entropy-coded using a factorized prior $P_{\hat{\boldsymbol{z}}}$. The complete optimization objective integrates both the main latent rate and the hyperprior side information rate:

$$\mathcal{L}_{RD} = \mathbb{E}\left[-\log P_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) - \log P_{\hat{\boldsymbol{z}}}(\hat{\boldsymbol{z}}) + \lambda \cdot D(\boldsymbol{x}, \hat{\boldsymbol{x}})\right] \tag{6}$$

By minimizing this loss via gradient descent, the system jointly learns the transform parameters and the probability models in an end-to-end manner.

Building upon this foundational framework, we introduce SAAF with Sparse Attention and Adaptive Frequency. The overall architecture is illustrated in Fig. 1. SAAF consists of three novel and synergistically integrated primary components: Sparse Attention Block, Adaptive Frequency Block, and Denoising-as-Regularizer module.

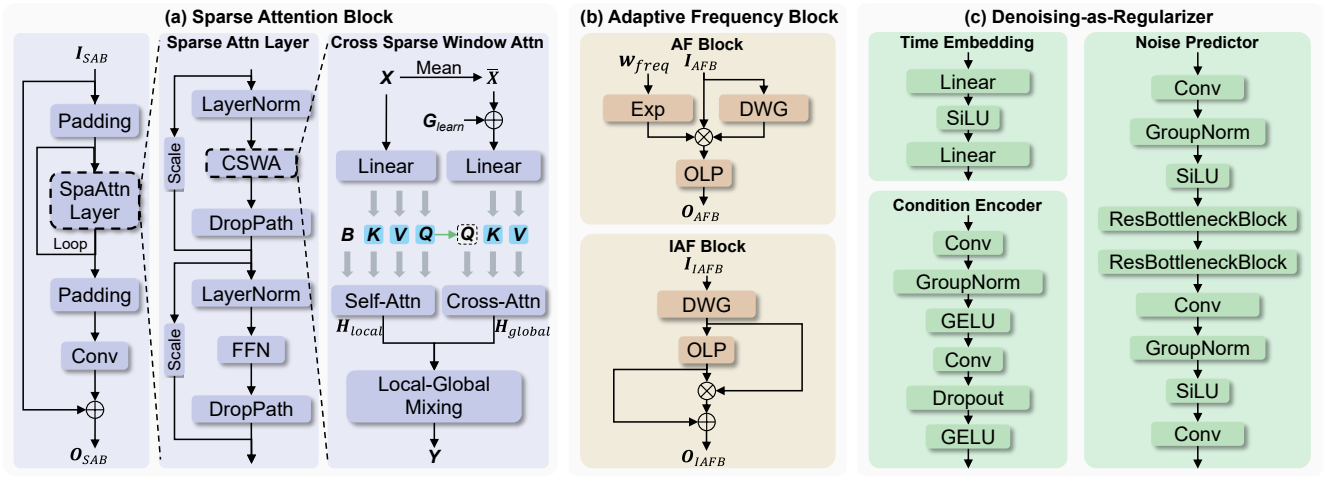

Figure 2. Detailed architectures of the Sparse Attention Block, Adaptive Frequency Block, and Denoising-as-Regularizer, respectively. In the SAB, we use Layer Normalization [4] as the feature layer. In the AFB, DWG represents Decomposition Weight Generator. In the DaR module, we employ GELU [20] and SiLU [40] as activation functions, Group Normalization [46] as the feature normalization layer, and the Residual Bottleneck Block [19] as the core feature extraction backbone of the Noise Predictor.

In the following sections, we will describe each of these components in detail, referencing the architectural diagrams in Fig. 2.

3.2. Sparse Attention Block (SAB)

While standard Window-based Multi-head Self-attention (WMSA) [28] achieves computational efficiency, its limited receptive field struggles with long-range dependencies. Swin Transformer's Shifted Window mechanism, subsequently adopted by some LIC methods [24, 25, 27, 29], addresses this limitation but introduces higher complexity by requiring multiple layers to propagate information across distant regions. Consequently, we propose Cross-Sparse Window Attention (CSWA), a more efficient local-global attention mechanism designed to supersede the standard WMSA module, enabling the effective modeling of spatial features. Proposed CSWA consists of three core parts: Local Window Attention (LWA), Global Sparse Attention (GSA), and Local-Global Mixing (LGM).

Given an input feature map $\boldsymbol{X} \in \mathbb{R}^{H\times W\times C}$, we partition it into $n_w = \frac{H}{M} \times \frac{W}{M}$ non-overlapping windows of size $M \times M$. For computation, each window is flattened into a token sequence. Since all windows are processed in parallel, we describe the computation for a single window $\boldsymbol{X}_i \in \mathbb{R}^{M^2\times C}, i = 1, ..., n_w$ for simplicity of notation.

3.2.1. Local Window Attention (LWA)

For window $\boldsymbol{X}_i \in \mathbb{R}^{M^2\times C}$, We compute queries, keys, and values via shared projections $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{C\times C}$:

$$\boldsymbol{Q}_i = \boldsymbol{X}_i\boldsymbol{W}_Q, \quad \boldsymbol{K}_i = \boldsymbol{X}_i\boldsymbol{W}_K, \quad \boldsymbol{V}_i = \boldsymbol{X}_i\boldsymbol{W}_V \in \mathbb{R}^{M^2\times C} \tag{7}$$

For multi-head attention with h heads, we reshape $\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i$ to $\{\boldsymbol{Q}_i^{(j)}, \boldsymbol{K}_i^{(j)}, \boldsymbol{V}_i^{(j)} \in \mathbb{R}^{M^2\times d_h}\}_{j=1}^{h}$, where $d_h = C/h$. The local attention output of j-th head is:

$$\boldsymbol{H}_{local,i}^{(j)} = \text{Softmax}\left(\frac{\boldsymbol{Q}_i^{(j)}(\boldsymbol{K}_i^{(j)})^\top}{\sqrt{d_h}} + \boldsymbol{B}\right)\boldsymbol{V}_i^{(j)} \in \mathbb{R}^{M^2\times d_h} \tag{8}$$

where $\boldsymbol{B} \in \mathbb{R}^{M^2\times M^2}$ is the relative position bias [28]. Crucially, to minimize inference latency, we diverge from the original Swin Transformer [28], which generates $\boldsymbol{B}$ dynamically via a Multi-Layer Perceptron (MLP). Instead, we precompute and cache $\boldsymbol{B}$ as a static look-up table, completely eliminating this computational overhead during inference. This optimization is a key contributor to our method's high efficiency, as quantitatively validated in Tab. 3.

3.2.2. Global Sparse Attention (GSA)

To capture cross-window dependencies efficiently, we introduce N_g learnable global tokens, denoted as $\boldsymbol{G}_i \in \mathbb{R}^{N_g\times C}$, which are window-conditioned to adapt to local context:

$$\boldsymbol{G}_i = \boldsymbol{G}_{learn} + \bar{\boldsymbol{X}}_i \tag{9}$$

where $\boldsymbol{G}_{learn} \in \mathbb{R}^{N_g\times C}$ are shared learnable parameters across all windows, and $\bar{\boldsymbol{X}}_i = \frac{1}{M^2}\sum_{k=1}^{M^2}\boldsymbol{X}_i^{(k)} \in \mathbb{R}^C$ is the mean spatial feature of window $\boldsymbol{X}_i$. This design allows each window to have context-specific global representations while maintaining high parameter efficiency.

We compute keys and values for the global tokens via shared projections $\boldsymbol{W}_g^{(K)}, \boldsymbol{W}_g^{(V)} \in \mathbb{R}^{C\times C}$:

$$\boldsymbol{K}_{g,i} = \boldsymbol{G}_i\boldsymbol{W}_g^{(K)} \in \mathbb{R}^{N_g\times C}, \quad \boldsymbol{V}_{g,i} = \boldsymbol{G}_i\boldsymbol{W}_g^{(V)} \in \mathbb{R}^{N_g\times C} \tag{10}$$

After reshaping to h heads ($\boldsymbol{K}_{g,i}^{(j)}, \boldsymbol{V}_{g,i}^{(j)} \in \mathbb{R}^{N_g\times d_h}, j = 1, ..., h$), we compute cross-attention between local queries

and global keys:

$$\boldsymbol{H}_{global,i}^{(j)} = \text{Softmax}\left(\frac{\boldsymbol{Q}_i^{(j)}(\boldsymbol{K}_{g,i}^{(j)})^\top}{\sqrt{d_h}}\right)\boldsymbol{V}_{g,i}^{(j)} \in \mathbb{R}^{M^2\times d_h} \tag{11}$$

Note that each of the M^2 local tokens attends to only N_g global tokens, resulting in a $M^2 \times N_g$ attention matrix (compared to $M^2 \times M^2$ in LWA).

3.2.3. Local-Global Mixing (LGM)

We fuse local and global outputs with weight α:

$$\boldsymbol{H}_i^{(j)} = (1-\alpha)\boldsymbol{H}_{local,i}^{(j)} + \alpha\boldsymbol{H}_{global,i}^{(j)} \tag{12}$$

where α is the weight factor. In this paper, we set α to a fixed value of 0.25. The outputs from all heads are concatenated and projected:

$$\boldsymbol{Y}_i = \text{Concat}(\boldsymbol{H}_i^{(1)}, \ldots, \boldsymbol{H}_i^{(h)})\boldsymbol{W}_{concat} \in \mathbb{R}^{M^2\times C} \tag{13}$$

All n_w windows are processed in parallel, then merged to form $\boldsymbol{Y} \in \mathbb{R}^{H\times W\times C}$. Finally, leveraging an advanced Transformer network [29] as the backbone, we incorporate our proposed CSWA to establish a robust spatial transform module, termed the Sparse Attention Block. As illustrated in Fig. 2 (a), this block effectively captures local and global spatial features to generate $\boldsymbol{O}_{SAB}$.

3.3. Adaptive Frequency Block (AFB)

Natural images contain multi-scale frequency patterns which are important for compression. However, most existing LIC methods neglect frequency patterns, focusing primarily on spatial transforms. A few recent works [14, 25] have introduced frequency transforms to optimize RD performance. However, much like traditional algorithms, these transforms still rely on fixed, human-designed parameters (e.g., a fixed wavelet basis), and thus lack content-adaptive capabilities. Therefore, to effectively overcome this limitation, we introduce the Adaptive Frequency Block (AFB) for content-adaptive frequency re-weighting. Instead of relying on fixed transforms, the AFB employs our Decomposition Weight Generator (DWG) to dynamically generate weight maps based on input content, simulating responses across different frequency bands. The block also incorporates Orthogonal Linear Projection (OLP) [25] to ensure transformation stability. As shown in Fig. 1, AFB and its counterpart, the Inverse AFB (IAFB), are employed in the analysis and synthesis transform, respectively.

3.3.1. AFB (Analysis Transform)

For an input feature map $\boldsymbol{X} \in \mathbb{R}^{H\times W\times C}$, the AFB involves three key stages:

Decomposition Weight Generator. The module employs a lightweight convolutional network (termed the Decomposition Weight Generator) to dynamically generate four weight maps, $\boldsymbol{A}_{freq} \in \mathbb{R}^{H\times W\times 4}$. These maps simulate the response to 4 different frequency bands (LL, LH, HL, and HH) based on the input content.

Learnable Band-wise Re-weighting. Instead of a hard decomposition, we perform a content-adaptive re-weighting of the input features. To provide a global frequency preference, we introduce learnable weights $w_{freq} \in \mathbb{R}^4$. The final re-weighted feature $\boldsymbol{X}_{freq} \in \mathbb{R}^{H\times W\times C}$ is computed as a weighted sum, where the input $\boldsymbol{X}$ is modulated by both the dynamic attention maps $\boldsymbol{A}_{freq}$ and the global weights w_{freq} to obtain the refined features:

$$\boldsymbol{X}_{freq} = \boldsymbol{X} \odot \left(\sum_{i=1}^{4} \boldsymbol{A}_{freq,i} \cdot \exp(\boldsymbol{w}_{freq,i})\right) \tag{14}$$

where $\odot$ denotes element-wise multiplication and the $\exp(\cdot)$ operation ensures the weights are positive. This design allows the model to dynamically adjust its frequency response based on both local content (via $\boldsymbol{A}_{freq}$) and a global prior (via $\boldsymbol{w}_{freq}$), to enable precise feature modulation.

Orthogonal Linear Projection. Finally, the reweighted features are passed through an Orthogonal Linear Projection (OLP) layer [25], which performs a channel-wise transformation, while an orthogonality constraint on its weights ensures training stability and information preservation.

3.3.2. IAFB (Synthesis Transform)

In the decoder, the Inverse Adaptive Frequency Block (IAFB) adopts a symmetric architecture. It first uses an OLP layer with an orthogonality constraint to restore the feature dimension. Subsequently, it employs a similar frequency attention mechanism to apply a residual enhancement to the features, selectively restoring image details during reconstruction. This symmetric design ensures a consistent, efficient, and stable handling of feature frequencies throughout the encoding and decoding process. Furthermore, our overall framework adopts a spatial-frequency dual-path combination manner similar to AuxT [25].

3.4. Denoising-as-Regularizer

Traditional rate-distortion optimization, $\mathcal{L}_{RD} = \mathbb{E}[R(\hat{\boldsymbol{y}}) + \lambda D(\boldsymbol{x}, \hat{\boldsymbol{x}})]$, lacks latent constraints, often causing visual artifacts. We propose the Denoising-as-Regularizer (DaR) (Fig. 1 (c)), a training-only module to structure the latent space $\boldsymbol{y}$. DaR performs a single-step conditional denoising task. It corrupts $\boldsymbol{y}$ with scaled Gaussian noise $\boldsymbol{\epsilon}$ to get $\boldsymbol{y}_{noise} = \boldsymbol{y} + t\cdot\boldsymbol{\epsilon}$. A lightweight noise predictor $f_{denoise}(\cdot)$ is trained to predict $\boldsymbol{\epsilon}$, conditioned on the time step t (as $\boldsymbol{t}_{emb}$) and the hyper-latent $\hat{\boldsymbol{z}}$ (as $\hat{\boldsymbol{z}}_{cond}$). The module is optimized via the $\mathcal{L}_{DaR}$ loss defined as:

$$\mathcal{L}_{DaR} = \mathbb{E}\left[||f_{denoise}(\boldsymbol{y}_{noisy}, \boldsymbol{t}_{emb}, \hat{\boldsymbol{z}}_{cond}) - \boldsymbol{\epsilon}||_2^2\right] \tag{15}$$

Grounded in denoising score matching, minimizing $\mathcal{L}_{DaR}$ maximizes the conditional log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{c})$ (where

Table 1. BD-rates (over VTM-9.1) and model efficiency of LIC methods.

Method	Venue	BD-rate over VTM-9.1 (%)↓			Latency↓	FLOPs↓	Params↓
		Kodak	CLIC	Tecnick	(ms)	(G)	(M)
ELIC	CVPR'22	-7.06	-	-	-	-	-
MLIC+	MM'23	-13.15	-13.98	-16.03	-	-	-
MLIC++	NCW ICML'23	-15.07	-14.46	-17.19	211	494	116
QARV	TPAMI'23	-5.83	-	-3.92	-	-	-
TCM	CVPR'23	-11.76	-9.16	-10.89	133	701	76
CCA	NIPS'24	-13.78	-11.90	-13.96	102	616	65
FLIC	ICLR'24	-14.60	-10.79	-14.37	119	245	70
WeConvene	ECCV'24	-8.37	-6.87	-7.80	132	905	106
AuxT	ICLR'25	-10.17	-9.38	-9.98	82	**216**	**46**
DCAE	CVPR'25	-17.00	-16.98	-20.11	74	427	119
LALIC	CVPR'25	-15.30	-15.42	-17.61	-	-	-
MambaIC	CVPR'25	-14.47	-	-	-	-	-
SAAF (Ours)	-	**-17.40**	**-17.35**	**-20.57**	**67**	434	123

NOTE: Bold text indicates the best results. Latency and FLOPs are measured on Kodak.

$\boldsymbol{c} = \{t, \hat{\boldsymbol{z}}\}$), enforcing a learnable prior. The $\hat{\boldsymbol{z}}$ condition enables spatial adaptivity: smooth regions are strongly regularized while textures are preserved, achieving implicit frequency-adaptive regularization. The total training objective combines the RD loss, the orthogonality loss ($\mathcal{L}_{OLP}$), and the regularizer loss ($\mathcal{L}_{DaR}$):

$$\mathcal{L} = \mathbb{E}[\mathcal{L}_{RD} + \lambda_{\text{OLP}}\mathcal{L}_{\text{OLP}} + \lambda_{\text{DaR}}\mathcal{L}_{\text{DaR}}] \tag{16}$$

where $\lambda_{\text{OLP}} = 0.1$ and $\lambda_{\text{DaR}} = 0.01$. Crucially, DaR is deactivated during inference, imposing zero additional computational overhead. Its gradients effectively regularize the encoder to produce smoother latents, thereby improving visual quality while maintaining bitrate efficiency.

4. Experiments

4.1. Setup

Datasets. Consistent with most existing works [16, 27, 29, 52], we selected the first 300k images from the OpenImages dataset [23] with a short-side dimension of at least 256 for model training. For comprehensive performance testing, we used three gold-standard datasets Kodak [22] (24 images with a size of 768 × 512), CLIC Professional Validation [11] (41 images with 2k resolution), and Tecnick [3] (100 images with a size of 1200 × 1200).

Baselines. We compared our SAAF with recent advanced LIC methods, including MLIC++ [21], TCM-large [27], CCA [16], FLIC [24], WeConvene [14], AuxT [25], DCAE [29], ELIC [18], MLIC+ [21], QARV [12], LALIC [13], and MambaIC [50]. Among these, we successfully reproduced the first 7 methods to compare the model efficiency and reconstruction details on our device.

Training Setting. Following the same parameter settings as existing works, during the training phase, we randomly cropped the training images into 256 × 256 patches and fed the data into the model with a batch size of 16. The number of epochs was set to 100, with an initial learning rate of 0.0001, which was decayed to 0.00001 at the 80th epoch. We optimized the model using MSE as the loss function and trained 6 models by setting the Lagrangian multiplier λ to 0.05, 0.025, 0.013, 0.0067, 0.0035, and 0.0018, respectively. All experiments were conducted on the NVIDIA RTX 6000 Ada Generation GPU (48 GB).

4.2. Results

4.2.1. Overall Performance Comparison

We conducted a comprehensive comparison of SAAF against baselines. The results shown in Tab. 1 reveal that SAAF achieves the optimal BD-rates across all three datasets, while also attaining the best latency (67 ms). MLIC++ achieves competitive RD performance, but suffers from the highest latency (211 ms), rendering it impractical for many applications. TCM, CCA, and FLIC represent a moderate compromise, with latencies ranging from 102 ms to 133 ms. Notably, although AuxT builds upon the smaller TCM-small, its RD performance is comparable to that of TCM-large, while also achieving faster inference. LALIC and MambaIC demonstrate highly competitive performance by leveraging the advanced RWKV and Mamba models, respectively, as their transform backbones. DCAE introduces a dictionary-based entropy model which significantly boosts RD performance (e.g., -17.00% on Kodak). Our proposed SAAF further introduces a novel dual-path transform network featuring Sparse Attention and Adaptive Frequency, achieving optimal RD performance and the lowest inference latency. Critically, this high efficiency is attained while maintaining FLOPs and Parameters comparable to the strong DCAE baseline. We present the RD curves for all baselines in Fig. 3. The results show that the proposed SAAF outperforms competing algorithms on the

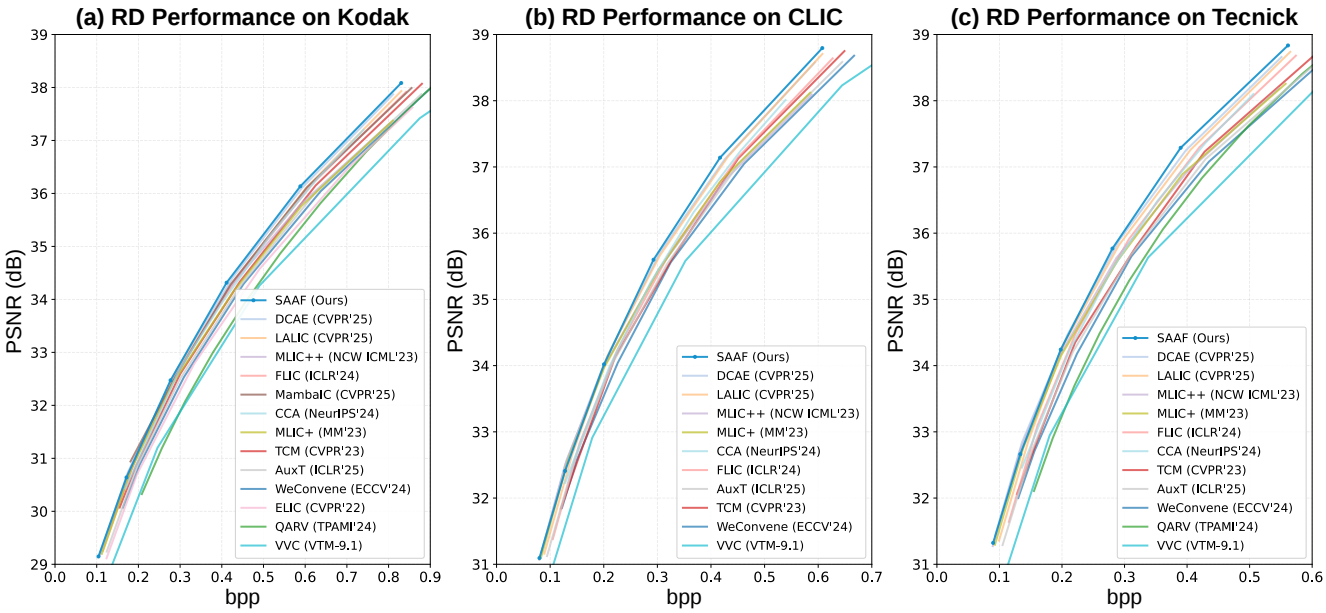

Figure 3. PSNR-bpp curves of LIC methods on Kodak, CLIC, and Tecnick datasets.

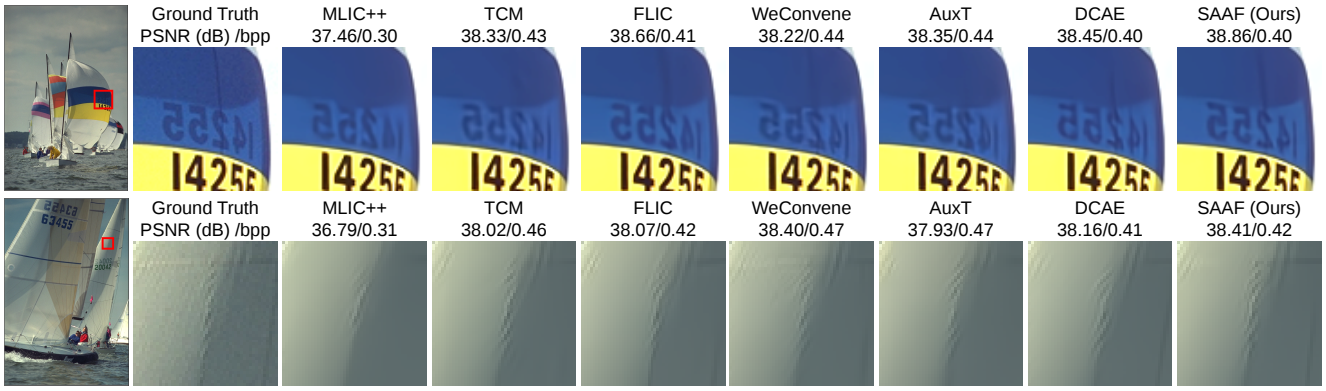

Figure 4. Visual comparison of reconstruction quality. The brightness is uniformly increased for clearer visualization.

Kodak, CLIC, and Tecnick datasets. Moreover, we compare the reconstruction quality between SAAF and baselines in Fig. 4. The visual details clearly demonstrate that SAAF achieves higher-quality reconstruction results, higher PSNR, and comparable bpp compared to competing methods.

4.3. Ablation Studies

For the ablation studies, to improve efficiency, we trained the models using only the first 30k images from the OpenImages dataset [23]. Furthermore, the three SABs in the analysis/synthesis network were configured with only 1, 2, and 4 Sparse Attention Layers (shown in Fig. 2 (a)), respectively. We designate the basic framework (using WMSA and no additional modules) as the BASE for all subsequent comparisons to quantify the performance gains.

4.3.1. Performance of Proposed Modules

We individually add the proposed SAB, AFB, and DaR modules to the BASE, and evaluate the performance of each configuration based on BD-rates over VTM-9.1 (on the Kodak, CLIC, and Tecnick datasets) and Latency (on the Kodak dataset). As shown in Tab. 2, the SAB (BASE+SAB), which replaces the standard WMSA in the BASE framework with CSWA, improves BD-rates while significantly reducing latency compared to the BASE. The AFB (BASE+AFB) achieves substantial BD-rate optimization with only a minor increase in latency. Furthermore, the training-only DaR module (BASE+DaR) effectively optimizes BD-rates at zero additional latency cost. Finally, the fully integrated model (BASE+ALL) achieves the optimal BD-rates and a highly competitive latency, a performance markedly superior to the BASE that demonstrates the effec-

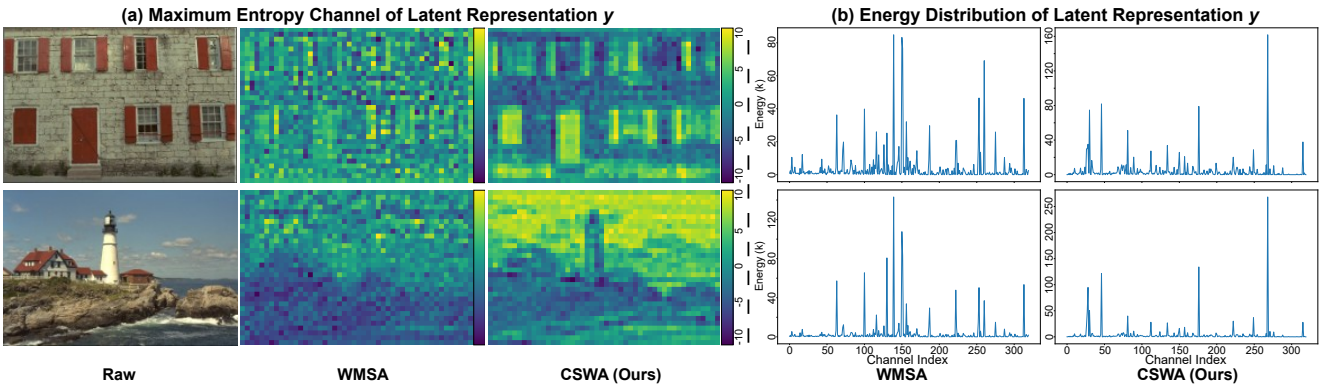

Figure 5. Visual analysis of the latent representation y obtained by WMSA and the proposed CSWA, respectively.

Table 2. BD-rates and latency of proposed modules.

BASE +	BD-rate over VTM-9.1 (%)↓			Latency (ms)↓
	Kodak	CLIC	Tecnick	(on Kodak)
-	-0.64	-1.86	-3.52	61
SAB ($N_g = 2$)	-1.86	-2.53	-4.16	**52**
AFB	-3.42	-4.35	-5.81	65
DaR	-1.56	-2.63	-4.29	61
ALL (SAAF)	**-3.99**	**-4.59**	**-6.04**	56
SAB ($N_g = 1$)	-1.32	-2.15	-3.54	51
SAB ($N_g = 3$)	-1.64	-2.33	-3.68	53

NOTE: ALL denotes the BASE combined with all proposed modules: SAB, AFB, and DaR.

tiveness of our proposed modules.

4.3.2. Ablation Study on the Number of Global Token

We investigate the effect of the number of learnable global tokens (G_{learn}) within the CSWA module on overall model performance. The results in Tab. 2 demonstrate that the model achieves the optimal BD-rate and a competitive latency when $N_g = 2$. This demonstrates that our proposed mechanism can achieve significant BD-rate optimization using an extremely sparse set of global tokens.

4.3.3. Comparison of WMSA and proposed CSWA

In addition to the RD performance and latency advantages demonstrated in Tab. 2 (comparing BASE and BASE + SAB), we further comprehensively analyze the feature extraction capabilities and module-level efficiency of the proposed CSWA compared to WMSA. As depicted in Fig. 5, we provide a visual analysis of the latent representation $\boldsymbol{y}$ generated by both approaches. Fig. 5 (a) visualizes the maximum entropy channel, revealing that CSWA captures the original image contours more clearly than WMSA. Furthermore, Fig. 5 (b) illustrates the energy distribution, indicating that CSWA excels at concentrating energy into a small subset of channels, thereby optimizing the compression for the subsequent entropy model. We also evaluated the computational efficiency of WMSA and our CSWA on identical feature maps. As shown in Tab. 3, CSWA achieves lower FLOPs, latency, and GPU memory usage, with only a slight increase in parameters compared to WMSA. This is attributed to CSWA's two key designs: introducing a sparse global token mechanism and replacing the dynamic MLP with a pre-computed relative position bias look-up table.

Table 3. Efficiency of WMSA and the proposed CSWA

Module	Latency↓ (ms)	FLOPs↓ (M)	Params↓ (M)	GMU↓ (MB)
WMSA	0.44	264.24	**0.15**	22.70
CSWA (Ours)	**0.33**	**231.21**	0.22	**21.82**

NOTE: GMU represents GPU Memory Usage.

5. Conclusion

In this paper, we propose an efficient LIC method to address the performance-latency challenge. We design a spatial-frequency dual-path transform network that integrates the Sparse Attention Block with CSWA for efficient long-range modeling and an Adaptive Frequency Block for content-adaptive frequency processing. Furthermore, we propose a Denoising-as-Regularizer module to improve RD performance at zero inference cost by leveraging the diffusion principle to regularize the latent space. Extensive experiments on standard datasets with different resolutions validate that our method outperforms existing state-of-the-art methods in both RD performance and inference latency.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant (62272252, 62272253) and the China Scholarship Council scholarship program.

References

[1] Mohammad Akbari, Jie Liang, Jingning Han, and Chengjie Tu. Generalized octave convolutions for learned multi-frequency image compression. *arXiv preprint arXiv:2002.10032*, 2020. 3

[2] Mohammad Akbari, Jie Liang, Jingning Han, and Chengjie Tu. Learned multi-resolution variable-rate image compression with octave-based residual blocks. *IEEE Transactions on Multimedia*, 23:3013–3021, 2021. 3

[3] Nicola Asuni and Andrea Giachetti. Testimages: A large-scale archive for testing visual devices and basic image processing algorithms. In *Proceedings of STAG*, pages 63–70, 2014. 6

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

[5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1, 2

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

[7] John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2, 1989. 1

[8] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 1

[9] Yunuo Chen, Zezheng Lyu, Bing He, Hongwei Hu, Qi Wang, Yuan Tian, Li Song, Wenjun Zhang, and Guo Lu. Cmic: Content-adaptive mamba for learned image compression. *arXiv preprint arXiv:2508.02192*, 2025. 1

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

[11] CLIC. Workshop and challenge on learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. CLIC (Workshop on Learned Image Compression), 2021. 6

[12] Zhihao Duan, Ming Lu, Jack Ma, Yuning Huang, Zhan Ma, and Fengqing Zhu. Qarv: Quantization-aware resnet vae for lossy image compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):436–450, 2023. 6

[13] Donghui Feng, Zhengxue Cheng, Shen Wang, Ronghua Wu, Hongwei Hu, Guo Lu, and Li Song. Linear attention modeling for learned image compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7623–7632, 2025. 1, 2, 6

[14] Haisheng Fu, Jie Liang, Zhenman Fang, Jingning Han, Feng Liang, and Guohe Zhang. Weconvene: Learned image compression with wavelet-domain convolution and entropy model. In *European Conference on Computer Vision*, pages 37–53. Springer, 2024. 2, 3, 5, 6

[15] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. 1, 2

[16] Minghao Han, Shiyin Jiang, Shengxi Li, Xin Deng, Mai Xu, Ce Zhu, and Shuhang Gu. Causal context adjustment loss for learned image compression. *Advances in Neural Information Processing Systems*, 37:133231–133253, 2024. 6

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

[18] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5718–5727, 2022. 1, 3, 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[20] D Hendrycks. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4, 1

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

[22] Eastman Kodak. Kodak lossless true color image suite (photocd pcd0992). Technical report, Eastman Kodak Company, 1993. 6

[23] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2(3):18, 2017. 6, 7

[24] Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Frequency-aware transformer for learned image compression. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3, 4, 6

[25] Han Li, Shaohui Li, Wenrui Dai, Maida Cao, Nuowen Kan, Chenglin Li, Junni Zou, and Hongkai Xiong. On disentangled training for nonlinear transform in learned image compression. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 3, 4, 5, 6

[26] Jianping Lin, Mohammad Akbari, Haisheng Fu, Qian Zhang, Shang Wang, Jie Liang, Dong Liu, Feng Liang, Guohe Zhang, and Chengjie Tu. Variable-rate multi-frequency image compression using modulated generalized octave convolution. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2020. 3

[27] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 14388–14397, 2023. 1, 2, 4, 6

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 4

[29] Jingbo Lu, Leheng Zhang, Xingyu Zhou, Mu Li, Wen Li, and Shuhang Gu. Learned image compression with dictionary-based entropy model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12850–12859, 2025. 2, 4, 5, 6

[30] Haichuan Ma, Dong Liu, Ning Yan, Houqiang Li, and Feng Wu. End-to-end optimized versatile image compression with wavelet-like transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1247–1263, 2020. 3

[31] Huidong Ma, Hui Sun, Liping Yi, Yanfeng Ding, Xiaoguang Liu, and Gang Wang. Msdzip: Universal lossless compression for multi-source data via stepwise-parallel and learning-based prediction. In *Proceedings of the ACM on Web Conference 2025*, pages 3543–3551, 2025. 1

[32] Huidong Ma, Hui Sun, Liping Yi, Xiaoguang Liu, and Gang Wang. Multi-source data lossless compression via parallel expansion mapping and xlstm. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 1

[33] Stephane Mallat. *A wavelet tour of signal processing*. Academic Press, 1999. 3

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

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

[36] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023. 2

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

[38] Shiyu Qin, Jinpeng Wang, Yimin Zhou, Bin Chen, Tianci Luo, Baoyi An, Tao Dai, Shutao Xia, and Yaowei Wang. Mambavc: Learned visual compression with selective state spaces. *arXiv preprint arXiv:2405.15413*, 2024. 1, 2

[39] Shiyu Qin, Jinpeng Wang, Yimin Zhou, Bin Chen, Tianci Luo, Baoyi An, Tao Dai, Shu-Tao Xia, and Yaowei Wang. Cassic: Towards content-adaptive state-space models for learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15727–15736, 2025. 1

[40] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 4

[41] Gabriele Spadaro, Alberto Presta, Enzo Tartaglione, Jhony H Giraldo, Marco Grangetto, and Attilio Fiandrotti. Gabic: Graph-based attention block for image compression. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 1802–1808. IEEE, 2024. 2

[42] Hui Sun, Huidong Ma, Feng Ling, Haonan Xie, Yongxia Sun, Liping Yi, Meng Yan, Cheng Zhong, Xiaoguang Liu, and Gang Wang. A survey and benchmark evaluation for neural-network-based lossless universal compressors toward multi-source data. *Frontiers of Computer Science*, 19(7):1–16, 2025. 1

[43] 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. *arXiv preprint arXiv:1511.06085*, 2015. 1

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, and ukasz Kaiser. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1

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

[46] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 4

[47] Zhuojie Wu, Heming Du, Shuyun Wang, Ming Lu, Haiyang Sun, Yandong Guo, and Xin Yu. Cmamba: Learned image compression with state space models. *arXiv preprint arXiv:2502.04988*, 2025. 1, 2

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

[49] Ali Zafari, Atefeh Khoshkhahtinat, Piyush Mehta, Mohammad Saeed Ebrahimi Saadabadi, Mohammad Akyash, and Nasser M Nasrabadi. Frequency disentangled features in neural image compression. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2815–2819. IEEE, 2023. 3

[50] Fanhu Zeng, Hao Tang, Yihua Shao, Siyu Chen, Ling Shao, and Yan Wang. Mambaic: State space models for high-performance learned image compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18041–18050, 2025. 1, 2, 6

[51] Xiaosu Zhu, Jingkuan Song, Lianli Gao, Feng Zheng, and Heng Tao Shen. Unified multivariate gaussian mixture for efficient neural image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17612–17621, 2022. 2

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