Rethinking Latent Variable in Learned Image Compression

Fangzhou Yi, Zhicheng Gong, Hui Zeng*
University of Science and Technology Beijing, Beijing 10083, China
{m202310569,m202310581}@xs.ustb.edu.cn
* Corresponding author hzeng@ustb.edu.cn

Abstract

Benefiting from the powerful data learning and representation capabilities of neural networks, Learned Image Compression (LIC) methods have demonstrated better Rate-Distortion (RD) performance than traditional image compression frameworks. In this paper, we analyze the role of latent variables in image compression, both qualitatively and quantitatively. We then propose a latent variable compensation method to mitigate the loss introduced by quantization. We also introduce a regularization term for the latent variable mean square error into the loss function, providing more explicit guidance for the compression and reconstruction of the model's internal latent variables. Additionally, we propose a noise compensation method that acts as a plug-and-play component to enhance reconstruction in lossy image compression without significant additional encoding or decoding time. We also present a data augmentation technique involving image inversion, which helps the training set conform to the symmetry inherent in probabilistic modeling for image compression tasks. Extensive experiments demonstrate that the proposed method enhances rate-distortion metrics and visual quality.

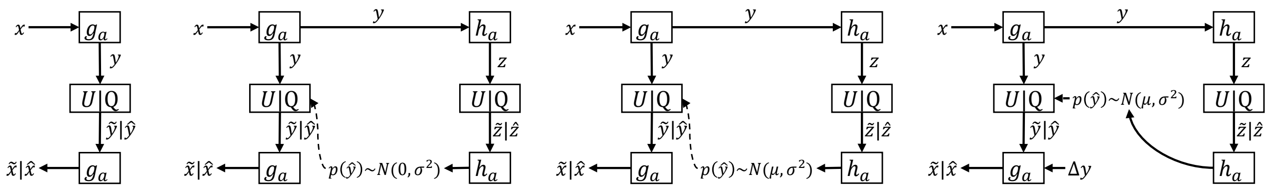
1. Introduction

In certain scientific fields such as medicine and remote sensing, high-precision, hyperspectral, high-resolution imaging is a basic requirement for research progress and breakthroughs. However, higher accuracy and resolution often results in larger image file sizes. For example, a high-resolution remote sensing image will take 3 GB space to store, while a remote sensing image dataset holds more than hundreds of mentioned ones, totally 3 TB. However, the storage capacity of today's computers is usually no more than 1 TB (1024 GB), and the bandwidth of network transmission is generally about 40 MB/s. It poses a great challenge to the storage and transmission of images. Compressing image data efficiently without losing much information has become an urgent problem that needs to be solved.

Early image compression was achieved mainly through cleverly designed transformations, and most of them were lossy compression. To make a balance between visual effect and compression performance, some information loss is acceptable, and storage space can be saved by removing unnecessary details. For example, the human eye is more sensitive to small changes in brightness than to changes in color. One of the most prominent of these works is the Discrete Cosine Transform (DCT), proposed by Nasir Ahmed [3], and later an algorithm for practical application was introduced by T. Natarajan and K.R. Rao. DCT express a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies.

In recent years, Variational Autoencoder [18] (VAE) based learned Image compression model Fig. 1 has achieved huge progress in rate-distortion performance on metrics of Signal-to-Noise Ratio (PSNR) and Multi-Scale-Structural Similarity Index Measure (MS-SSIM) [35]. This kind of architecture can be interpreted as VAEs based on transform coding. Through encoder $g_a(\cdot)$, raw images $\mathbf{x}$ are transformed to latent space as latent representation $\mathbf{y} = g_a(\mathbf{x})$, then quantized $\hat{\mathbf{y}} = Q(\mathbf{y})$ for practical codes, and entropy encoded to bit streams. In the decode stage, bit streams are decompressed to latent variables and then inversely transformed to reconstruct lossy images $\hat{\mathbf{x}} = g_s(\hat{\mathbf{y}})$. To achieve higher reconstruct and compression performance, the whole framework is optimized with PSNR or MS-SSIM and the entropy of the quantized latent variables.

For practical coding implementations, researchers prefer using integer symbols over floating-point symbols, necessitating quantization prior to encoding. It is important to note that this quantization process is applied exclusively during the inference phase. To facilitate the smooth backpropagation of gradients during training, researchers have developed various simulated quantization operations [2, 5]. However, this discrepancy in workflows between the training and inference phases introduces variations between the estimated and actual encoded bitrate, as well as the reconstructed images. While the Straight-Through Estimator (STE), introduced in [26], helps align the reconstructed im-



(a) Factorized (b) ScaleHyperprior (c) MeanScaleHyperprior (d) Proposed

Figure 1. Operational diagrams of learned compression models and proposed latent compensation model

ages between estimated and actual encoding, a gap in bitrate estimation still persists.

In this paper, we proposed a easy-to-implement noise compensation approach for lossy image compression. By learning distribution of latent residual, we achieve better rate-distortion performance. We also made attempts to improve and accelerate the training of LIC. The main contributions of this paper are summarized as follows:

- We introduce a regularization term—the latent variable mean square error—into the loss function to provide more explicit guidance for the compression and reconstruction of the model's internal latent variables.
- We propose a noise compensation method designed to enhance reconstruction in lossy image compression. This method acts as a plug-and-play component, improving existing models without requiring much additional encoding or decoding time.
- We present a data augmentation technique involving image inversion, which helps the training set better conform to the symmetry inherent in probabilistic modeling for image compression tasks.

2. Related Work

In this section, we present related work in two main areas. Firstly, we delve into learned image compression, an prosperous area of research. Secondly, we discuss the techniques for image data augmentation, which is the focus of this paper.

2.1. Learn Image Compression

Learning-based lossy image compression started with the work of Toderici et al. Toderici et al. [33] introduced long-short-term memory (LSTM) networks to image compression. Their method encodes images or the residuals of images progressively, allowing for multi-rate image compression. By increasing the number of recurrent neural network (RNN) iterations, they can enhance the compression efficiency and image quality. This technique leverages the temporal capabilities of LSTMs to effectively model dependencies in image data, leading to better performance compared to traditional compression methods. After that, Toderici et al. [34] and Johnston et al. [17] further modified the RNN structure, which introduced LSTM-based entropy coding and was more concerned with spatially adaptive bit allocation. Ballé [6] proposed a general end-to-end convolution neural network and started the age of VAEs as the backbone. This framework is followed by the most recent learned lossy image compression methods. There are three dominant kinds of improvements applied in the VAEs framework, including network architectures, quantization, and entropy encoding.

Network Architectures. Chen [10] integrated simplified attention modules into the network architecture. These modules focus on complex regions of the image while reducing the bits required for simpler regions, enhancing overall coding performance with moderate computational complexity. Ma [24] proposed iWave, a reversible wavelet-like transform that enables lossless conversion of images into coefficients. Unlike traditional auto-encoders, this transform avoids information loss and supports lossless and lossy compression. Zhu [40] leveraged Swin Transformers to develop non-linear transforms that replace traditional convolutional neural networks (CNNs). This approach results in improved compression efficiency and a better rate-distortion-computation trade-off.

Quantization. Agustsson [2] introduced a soft-to-hard quantization approach, which utilizes soft assignments during training to approximate quantization, enabling gradient-based optimization. Gradually this approach annealed to hard assignments for practical use, ensuring an efficient transition from smooth to discrete representations. Menzter [25] used a lightweight 3D convolutional neural network as a context model to estimate the entropy of the latent representation and deployed a differentiable soft quantization approach ensures smooth gradient flow during backpropagation. Li [22] introduced an importance map subnet to produce a locally adaptive bit-rate allocation mask, which effectively allocates more bits to image regions with complex structures while reducing bits for smoother areas. The authors also developed a trimmed convolutional network to predict the conditional probability of quantized codes, enabling efficient large-context modeling. Xiang [36] integrated a quantization loss compensation (QLC) network into the network architecture, learning compensatory information for details lost during quantization. Yang [38] proposed a Stochastic Gumbel Annealing (SGA) method to narrow the discretization gap before and after quantization.

Entropy Encoding. Ballé [5] extended the traditional autoencoder-based image compression by integrating a hyperprior that effectively models spatial dependencies in the latent representation, enabling the model to better capture local variations in image statistics. Qian [31] introduced a global reference model that allows for better representation of image content in entropy coding.The global reference improves the model's ability to predict and code residuals, the differences between the predicted and actual pixel values, more accurately, resulting in improved compression performance. Minnen [27] combined autoregressive model with hierarchical priors and leveraged their complementary strengths, achieving better exploitation of probabilistic structures in latent representations. Aforementioned methods slowed down the coding operation to some extent. He [13] introduced a checkerboard-shaped spatial context model. Unlike previous models, this new method enables parallel decoding, significantly enhancing computational efficiency. Minnen [26] generalized Gaussian Scale Mixtures (GSM) to Gaussian Mixture Models (GMM), improving the match between the entropy model and latent distributions. After this work, they further introduced a channel-wise autoregressive model, by splitting the latent tensor into multiple slices along the channel dimension, successfully reduced spatial redundancy and improving compression efficiency. Fu [11] further proposed a more flexible discretized Gaussian-Laplacian-Logistic mixture model (GLLMM) for the latent representations. Based on existing channel spliting methods, Hu [16] implemented an unevenly grouped channel-conditional coding strategy. By allocating more channels to earlier groups (with more significant information) and fewer to later groups, this strategy further reduced computational complexity while maintaining high RD performance.

3. Method

3.1. Preliminary

Since our method is built upon the hyper-prior architecture, we would briefly introduce the basic pipeline for better understanding. Assuming that $\mathbf{x}$ are raw images sampled from a probability distribution $p(\mathbf{x})$, the smallest average code length of the compressed bit-stream is the theoretical Shannon entropy:

$$H(p) = \mathbb{E}_{p(\mathbf{x})}[-\log p(\mathbf{x})] \tag{1}$$

we deploy a estimate probability distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ to represent the real probability distribution $p(\mathbf{x})$, where $\boldsymbol{\theta}$ is the parameters of this model. The approximate compression performance can be calculated by the cross entropy of the two probability distributions:

$$H(p, p_{\boldsymbol{\theta}}) = \mathbb{E}_{p(\mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x})] \geq H(p) \tag{2}$$

And only when $p(\mathbf{x})$ equals to $p_{\boldsymbol{\theta}}(\mathbf{x})$, it becomes an equation.

It is often infeasible to directly compress the original variable x using an estimated distribution, which would bring in extremely computation cost. So we introduce a high-dimension latent variable y to achieve more efficient estimation. The estimate probability distribution can be written as a marginal distribution:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})d\mathbf{y} = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})p_{\boldsymbol{\theta}}(\mathbf{y})d\mathbf{y} \tag{3}$$

In (3) , we only have the inverse direction from $\mathbf{y}$ to $\mathbf{x}$. Following Bayes rule, we rewrite the equation intuitively:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}}(\mathbf{x}) \Leftrightarrow p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})}{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})} \tag{4}$$

Now we get the forward direction from $\mathbf{x}$ to $\mathbf{y}$, which indicates the transformation from low dimension variable $\mathbf{x}$ to high-dimension latent variable $\mathbf{y}$.

By introducing an inference model $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$, the logarithm of the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{x})$ can be written as:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_{\phi}(\mathbf{y}|\mathbf{x})} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})}{q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})}}_{ELBO} + \underbrace{\mathbb{E}_{q_{\phi}(\mathbf{y}|\mathbf{x})} \log \frac{q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}}_{D_{kl}(q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})|p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}))} \tag{5}$$

where $D_{kl}(q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})|p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}))$ is the Kullback-Leiber (KL) divergence, which measures the difference between two distributions. Because the $D_{kl}(q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})|p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})) \geq 0$, ELBO becomes the lower bound of $\log p_{\boldsymbol{\theta}}(\mathbf{x})$:

$$\mathbb{E}_{p(\mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x})] \leq \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q_{\phi}(\mathbf{y}|\mathbf{x})}[-\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})}{q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})}] \tag{6}$$

and can minimized the expectation of negative ELBO as a proximity for expected code length $\mathbb{E}_{p(\mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x})]$.

The optimization problem can be represented as a variational autoencoder:

$$\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q_{\phi}(\hat{\mathbf{y}}|\mathbf{x})}\left[\cancelto{0}{\log q_{\boldsymbol{\phi}}(\hat{\mathbf{y}}|\mathbf{x})} \underbrace{-\log p_{\boldsymbol{\theta}}(\mathbf{x}|\hat{\mathbf{y}})}_{WeightDistortion} \underbrace{-\log p_{\boldsymbol{\theta}}(\hat{\mathbf{y}})}_{Bitrate}\right] \tag{7}$$

As introduced in [5], the quantization operation can be relaxed by adding uniform noise, and assume $q_{\boldsymbol{\phi}}(\hat{\mathbf{y}}|\mathbf{x}) =$

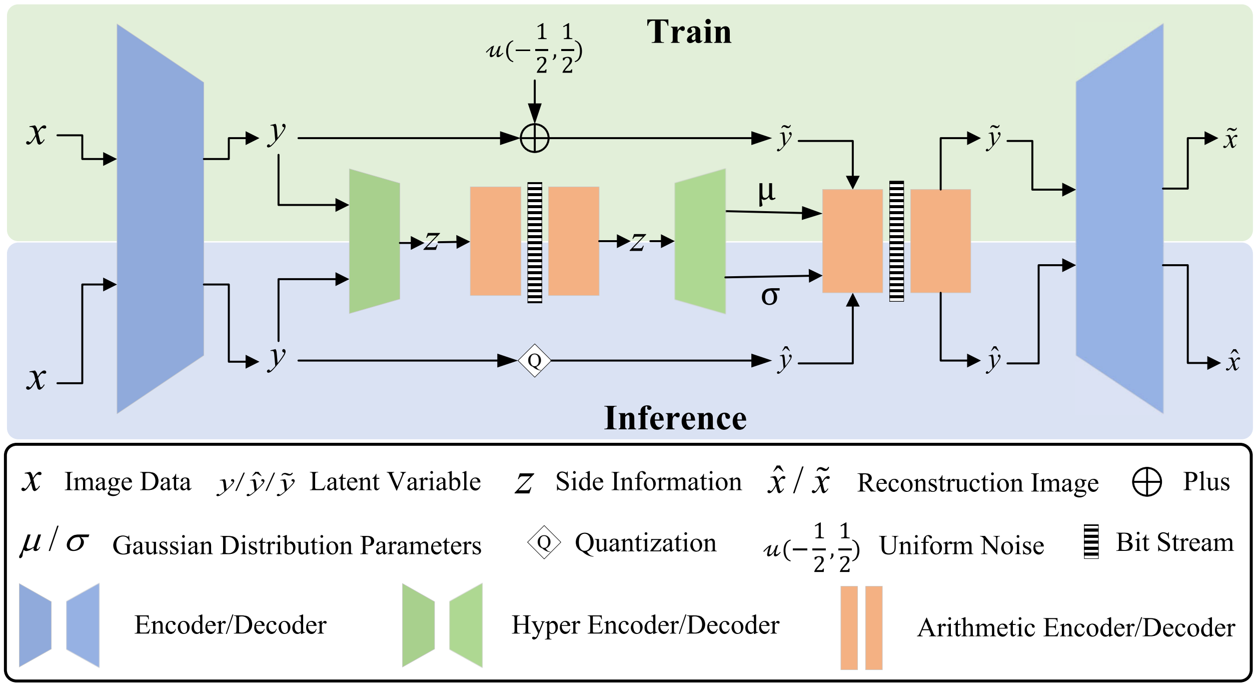

Train
Inference
$\mathcal{U}(-\frac{1}{2},\frac{1}{2})$
x
y
$\tilde{y}$
$\hat{y}$
z
μ
σ
$\tilde{x}$
$\hat{x}$
Q
x Image Data
$y/\hat{y}/\tilde{y}$ Latent Variable
z Side Information
$\hat{x}$ / $\tilde{x}$ Reconstruction Image
Plus
μ / σ Gaussian Distribution Parameters
Quantization
$\mathcal{U}(-\frac{1}{2},\frac{1}{2})$ Uniform Noise
Bit Stream
Encoder/Decoder
Hyper Encoder/Decoder
Arithmetic Encoder/Decoder

Figure 2. Two phases of compression. Training phase add latent variable $\mathbf{y}$ with uniform noise $\mathcal{U}(-\frac{1}{2},\frac{1}{2})$ to establish bitrate estimation. The inference phase adopts quantization on latent variable $\mathbf{y}$ and directly obtains bitrate from $\hat{\mathbf{y}}$.

$\prod_i \mathcal{U}(y_i - \frac{1}{2}, y_i + \frac{1}{2})$. Thus, $\log q_{\boldsymbol{\phi}}(\hat{\mathbf{y}}|\mathbf{x}) = 0$. For single lossy image compression, the second term of Eq. (7) usually can be regarded as the distortion loss between $\mathbf{x}$ and its lossy reconstruction $\tilde{\mathbf{x}}$ derived from $\hat{\mathbf{y}}$. The third term in Eq. (7) can be regarded as the rate loss of lossy image compression. In lossy compressor, only $\hat{y}$ needs to be stored as bit streams.

By introducing side information $\hat{z}$ to capture the spatial dependencies, we can model the condition probability distribution $p_{\boldsymbol{\theta}}(\hat{y}|\hat{z})$ as a Gaussian distribution with its mean μ and deviation σ, $p_{\boldsymbol{\theta}}(\hat{y}|\hat{z}) = \mathcal{N}(\mu, \sigma^2)$, and the optimization problem can further extended as:

$$\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{p_{\boldsymbol{\theta}}(\hat{\mathbf{y}}|\mathbf{x})}\left[\underbrace{-\log p_{\boldsymbol{\theta}}(\mathbf{x}|\hat{\mathbf{y}})}_{WeightDistortion} \underbrace{-\log p_{\boldsymbol{\theta}}(\hat{\mathbf{y}}|\hat{\mathbf{z}}) \underbrace{-\log p_{\boldsymbol{\theta}}(\hat{\mathbf{z}})}_{SideInformation}}_{Bitrate}\right] \tag{8}$$

Thus the loss function of image compression can be write as:

$$\begin{aligned}\mathcal{L} =& R + \lambda \cdot D \\ =& \mathbb{E}_{p(\mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\hat{\mathbf{y}}|\hat{\mathbf{z}}) - \log p_{\boldsymbol{\theta}}(\hat{\mathbf{z}})] \\ &+ \lambda \cdot \mathbb{E}_{p(\mathbf{x})}[d(\mathbf{x}, \hat{\mathbf{x}})]\end{aligned} \tag{9}$$

where R is the bit-rate of compressed image bit stream and D is the reconstruction quality metrics (MSE, PSNR, MS-SSIM, etc.). λ is not only used as a trade-off between compression rate and reconstruction quality, but also can be regarded as alignment for two variables with different scales.

3.2. Motivation

Modeling the image compression process as a VAE successfully established the theoretical foundation for LIC. However, this modeling approach also introduced certain errors. The latent variable $\mathbf{y}$ in Eq. (6) is simplified as quantized one $\hat{\mathbf{y}}$. However, this process is inherently lossy, leading to the degradation or complete loss of fine-grained information embedded within $\mathbf{y}$.

Rewrite Loss Function. First term in Eq. (7) is the discrete estimation for the continue expectation and we can extend its original form futher:

Proposition 1. $\log q_\phi(y|x) = k \cdot \sum \log q_\phi(\hat{y}|y, x)$.

Proof. For $\log q_\phi(\mathbf{y}|\mathbf{x})$, we have $\log q_\phi(\mathbf{y}|\mathbf{x}) = \sum \log q_\phi(\mathbf{y}, \hat{\mathbf{y}}|\mathbf{x})$. $\log q_\phi(\mathbf{y}, \hat{\mathbf{y}}|\mathbf{x})$ can be reformulated as $\log q_\phi(\hat{\mathbf{y}}|\mathbf{y}, \mathbf{x}) + \log q_\phi(\mathbf{y}|\mathbf{x})$. The second term is independent of $\hat{\mathbf{y}}$, so we have

$$\log q_\phi(\mathbf{y}|\mathbf{x}) = \sum \log q_\phi(\hat{\mathbf{y}}|\mathbf{y}, \mathbf{x}) + k^* \cdot \log q_\phi(\mathbf{y}|\mathbf{x}),$$

where k^* is the number of possible values that $\hat{\mathbf{y}}$ can take. Then

$$\log q_\phi(\mathbf{y}|\mathbf{x}) = \frac{1}{1-k^*} \log q_\phi(\hat{\mathbf{y}}|\mathbf{y}, \mathbf{x}).$$

To simplify the coefficient $\frac{1}{1-k^*}$, we set $\frac{1}{1-k^*}$ as k. $\square$

Consider $\log q_\phi(\hat{\mathbf{y}}|\mathbf{y}, \mathbf{x})$ as the error introduced by the quantization of the model's internal latent variables and the whole loss function can be written as:

$$\begin{aligned}\mathcal{L} =& R + \lambda \cdot D \\ =& \mathbb{E}_{p(\mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\hat{\mathbf{y}}|\hat{\mathbf{z}}) - \log p_{\boldsymbol{\theta}}(\hat{\mathbf{z}})] \\ &+ \lambda_1 \cdot \mathbb{E}_{p(\mathbf{x})}[d(\mathbf{x}, \hat{\mathbf{x}})] + \lambda_1 \cdot \lambda_2 \cdot \mathbb{E}_{p(\mathbf{x})}[d(\mathbf{y}, \hat{\mathbf{y}})]\end{aligned} \quad (10)$$

3.3. Quantization Loss Compensation

As introduced in Sec. 1 and Fig. 2, the quantization of the continuous latent variable $\mathbf{y}$ into its discrete counterpart $\hat{\mathbf{y}}$ is an indispensable step for practical coding in LIC. Additionally, this information loss directly contributes to the observed discrepancy in performance between the training and inference phases. Specifically, during training, models may optimize against a potentially continuous or noisily quantized latent representation, while inference strictly relies on hard-quantized discrete values.

We define the quantization error as $\Delta\mathbf{y} = \mathbf{y} - \hat{\mathbf{y}}$. Our empirical analysis, exemplified by the distribution of $\Delta\mathbf{y}$ sampled from the Kodak dataset in Fig. 3, reveals that this error largely follows a symmetric distribution centered around $x = 0$. This statistical characteristic of quantization error, where the mean is approximately zero for properly designed quantizers and uniformly distributed noise, forms the basis of our compensation strategy.

Minimizing the variance of this distribution $D(\Delta\mathbf{y})$, intuitively corresponds to reducing the information loss caused by quantization, as it would ideally converge to a Dirac delta function for perfect reconstruction.

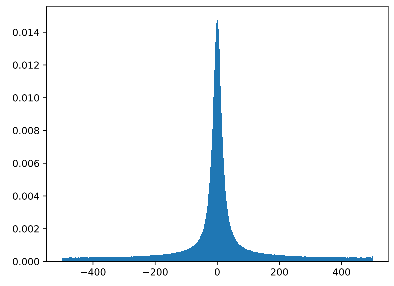


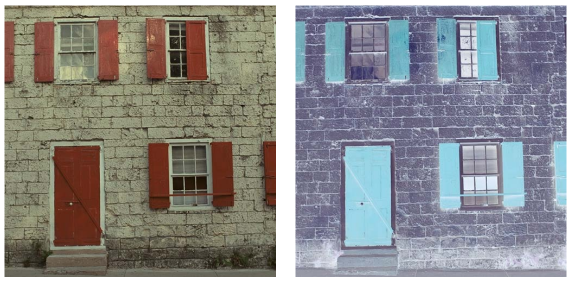
Figure 3. The distribution of $\Delta\mathbf{y}$ sampled from kodak dataset. We rescale the horizontal interval from [-0.5, 0.5] to [-500, 500]

Building upon this observation, we propose a noise compensation approach designed to mitigate the loss introduced by quantization. We explore two distinct methodologies for this compensation, Fixed-Parameter Compensation and Learned-Parameter Compensation. The detailed compensation implement will be provided in the supplementary material Fig. 7.

Fixed-Parameter Compensation (FPC). In this approach, the compensation noise e_{comp} is drawn from a predetermined distribution $p_e(\mu_{fixed}, \sigma^2_{fixed})$, where μ_{fixed} and σ^2_{fixed} are parameters derived from prior observation or theoretical analysis of typical quantization error characteristics. For instance, given the expectation of noise being zero, μ_{fixed} can be set to 0. The standard deviation σ_{fixed} might be empirically set based on a general understanding of $\Delta\mathbf{y}$ distribution (e.g., $3\sigma = \frac{1}{2}$ as hinted by typical Gaussian noise range for unit quantization interval), or from an analysis of representative datasets before training the full compression model. This method offers simplicity and guarantees zero bitrate overhead, as no parameters need to be learned or transmitted. However, it relies on a generalized statistical assumption which might not perfectly capture the context-dependent nature of quantization error.

Learned-Parameter Compensation (LPC). To overcome the limitations of fixed parameters, we also propose an approach where the compensation noise e_{comp} is drawn from a Gaussian distribution $\mathcal{N}(\mu_{\text{learn}}, \sigma^2_{\text{learn}})$, where μ_{learn} and σ_{learn} are learnable parameters integrated into the end-to-end optimization of the VAE. This means that these parameters are optimized jointly with the encoder (ϕ) and decoder ($\boldsymbol{\theta}$) to minimize the overall reconstruction error. Specifically, an auxiliary loss term (e.g., negative log-likelihood of observed errors $\mathbf{y} - \hat{\mathbf{y}}$ under $\mathcal{N}(\mu_{\text{learn}}, \sigma^2_{\text{learn}})$) encourages μ_{learn} and σ_{learn} to accurately represent the actual statistics of Δy during training. This ensures the learned distribution precisely reflects the true error characteristics from the dataset, adapting to the specific encoder-decoder architecture and data distribution.

3.4. Inverse Training



(a) Origin image (b) Inverse image

Figure 4. Demonstration image pair (kodim01.png from Kodak dataset)

Ballé [5, 27] models each element $\hat{y}_i$ in latent variable probablity with a Gaussian Distribution with its own mean μ_i and standard deviation σ_i. Many subsequent research [14, 23] methods have followed this principle. It is worthnoting that for each element $\hat{y}_i$, its distribution is symmetric about μ_i. Common data augmentation methods improve model performance and generalization by modifying the distribution of training data. Similarly, we can also employ certain data augmentation techniques to make the training data more suitable for image compression tasks. For acquired natural image datasets, we cannot fully guarantee their symmetry. For an RGB image, as show in Fig. 4, we can obtain its inverted image by applying $255. - \mathbf{x}$. It's clear that after inversion, the origin image and the inverted image achieve a certain symmetrical relationship at the pixel level.

Furthermore, we can label the origin image $\mathbf{x}$ like Fig. 4a and its inverted counterpart $\mathbf{x}^{'}$ like Fig. 4b as an image pair $\{\mathbf{x}, \mathbf{x}^{'}\}$. The original and inverted images maintain consistency in structural information. In lossless compression, we can recover either image from an image pair. For instance, after obtaining a lossless reconstructed image of the original $\mathbf{x}$ through a network, its inverted counterpart $\mathbf{x}^{'}$ can be derived by a simple subtraction operation. Similarly, we aim to achieve this in lossy compression as well.

4. Experiment

4.1. Experiment Settings

Training details. We train our network on 300k images randomly chosen from the OpenIamges dataset [20]. Before training, the images are first cropped into non-overlapped patches with the size of 256×256. We optimize the proposed network for 100 epochs (about 1.8M steps) using Adam with minibatches of size 16. The learning rate is initially set for 1×10^4 and is decayed by 0.3 at the epoch $\in [50, 75]$. Our models are all optimized for MSE, and the λ values belong to set $\{18, 35, 67, 130, 250, 483\} \times 10^{-4}$. Following the set of Ballé [5], the channel number of latent N and hyper latent M are chosen dependent on λ, with 192 and 128 for the lower 3 values, while 320 and 192 for the higher 3 values. The whole coding system is implemented with *Pytorch*, especially the *compressAI* [7] tool. We train the coding system on NVIDIA 2080ti GPU.

Evaluation datasets. We evaluate our network on the commonly used three image datasets.

- *Kodak.* [1] Kodak dataset [19] consists of 24 768×512 lossless true color images, which are widely used lossy image compression evaluation.
- *CLIC.m.* [2] CLIC mobile validation dataset [1] contains 61 different resolutions images taken with mobile phones.
- *CLIC.p.* [3] CLIC professional validation dataset [1] consists of 41 2K resolution images taken by professional photographers.

Evaluation metrics. We measure bit-stream length by bits per pixel(bpp), and measure reconstruction performance by pixel-level distortion metrics PSNR, MS-SSIM. All following EvalLoss is difined as the origin RD-Loss: $\mathcal{R} + \lambda_1 \cdot \mathcal{D}$. Some works [8, 9, 28–30] propose neither PSNR nor MS-SSIM match human perception. Additionally, we adopt reference perceptual metric LPIPS [39] and no-reference perceptual metrics FID [15].

Baseline methods and Comparison. The VAE based models share similar pipeline like Fig. 1. We integrate FPC into the fundamental scale-hyperprior model (Hyperprior) [5] and LPC into Mean-Scale-hyperprior model (Mbt-mean) [27]. We compare our method with Factorized Prior model (factorized) [6], Hyperprior [5], scale hyperprior with non zero-mean Gaussian conditionals (Mbt-mean) [27], joint auto-regressive hierarchical priors model (Mbt) [27], auto-regressive hyperprior model with convolutions (Anchor) [10], auto-regressive hyperprior model with GMM and simplified attention module (Attn) [10], auto-regressive hyperprior model with uneven channel groups with checkerboard spatial context (ELIC) [14].

4.2. RD Performance.

Fig. 5a - Fig. 5b show the comparison results on Kodak dataset. Results show that LPC module exploits extra average 0.2 dB boost in PSNR with similar PSNR performance and the quantifiable results in Sec. 4.3 indicate that LPC achieves 6.49% saving in BD-rate. For the hand-designed FPC module, its role is more akin to transforming the entire MSE-optimized model into one optimized based on MS-SSIM. As shown in Fig. 5a - Fig. 5b and Sec. 4.3, Hyperprior [5] with FPC module performs slightly worse than the baseline model in terms of PSNR, but achieves a significant improvement in MS-SSIM, even surpassing the Mbt-mean [27] which includes an LPC module. As illustrated in Fig. 5c - Fig. 5d, the comparison on the CLIC mobile and CLIC professional datasets indicate the same conclusion.

4.3. Codec Efficiency Analysis

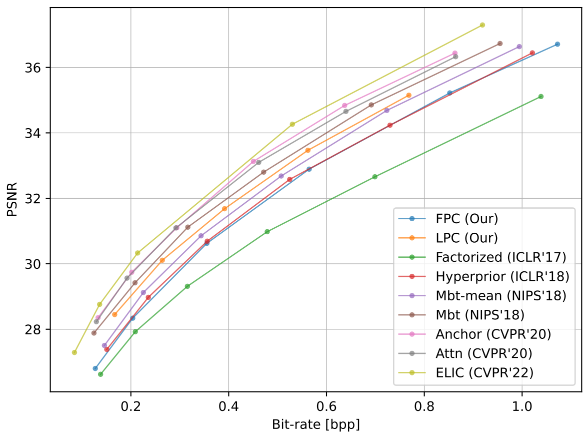
Method	Enc & Dec (s)	BD-Rate(%)↓	BD-MS-SSIM(dB)↑
Hyperprior	0.22	0.0	0.0
FPC	0.22	+0.79	**+0.30**
Mbt-mean	0.23	0.0	0.0
LPC	0.23	**-5.54**	+0.07

Table 1. Comparison of the averaged encoding and decoding time with Hyperprior [5] on Kodak dataset using GPUs (2080Ti).

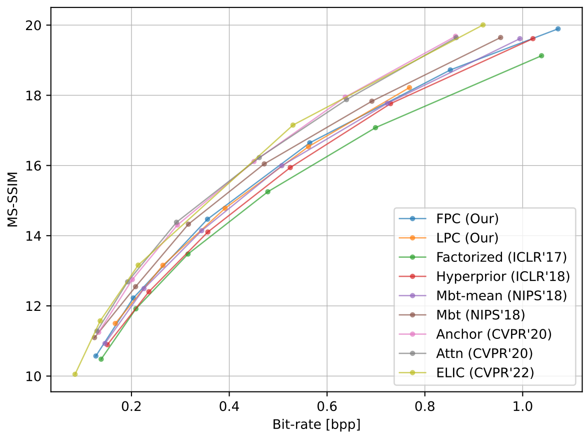
[1] https://r0k.us/graphics/kodak/

[2] https://data.vision.ee.ethz.ch/cvl/clic/mobile_valid_2020.zip

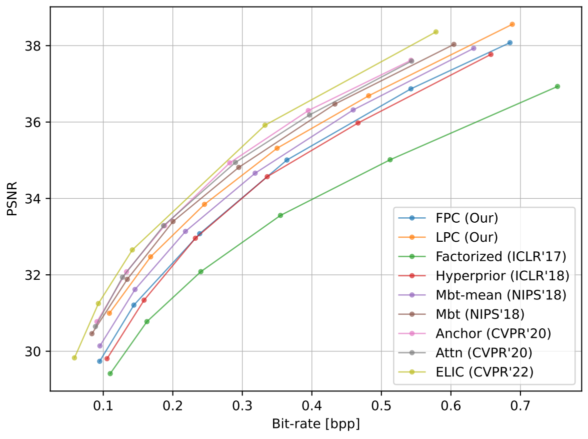
[3] https://data.vision.ee.ethz.ch/cvl/clic/professional_valid_2020.zip



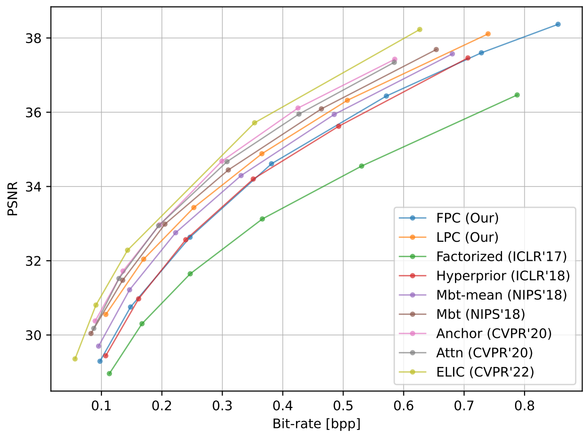
(a) PSNR on Kodak dataset



(b) MS-SSIM on Kodak dataset



(c) PSNR on CLIC.m. dataset



(d) PSNR on CLIC.p. dataset

Figure 5. Comparisons of RD curves with other methods on the Kodak, CLIC.m., and CLIC.p. datasets.

In Sec. 4.3, we evaluate the inference latency of our method as a plug-and-play module into Hyperprior [5] and Mbt-mean [27] model on the Kodak dataset. We set the baseline model as the reference value 0.For the FPC module, since it is hand-designed, it does not introduce any additional training parameters. For the LPC module, we introduced additional network connections to learn the distribution parameters μ, σ of the residuals. This does increase the overall number of model parameters, but the added amount is almost negligible compared to the total parameters of the baseline model. Consequently, the new module's impact on the model's computational efficiency is also negligible. We attach qualitative analysis for CLIC datasets in supplementary materials Tab. 6.

4.4. Visual Quality.

Fig. 6 show the example of reconstructed images (*kodim04.png*) by our methods LPC and the baseline method Mbt-mean [27]. Our reconstructed images retain more details with similar bitrate.

4.5. Ablation Study

Effects of Inverse Training. We select three numbers of dataset to quantify the effect of Inverse training. As shown in Tab. 2, for all tested original dataset sizes (300k, 150k, and 75k), Inverse Training trick consistently leads to a lower Eval Loss compared to not using inverse training. The inverse training trick not only ingeniously expands the training dataset but also significantly enhances the model's performance and generalization ability.

This trick, to some extent, also breaks the previous constraint in LIC model training, namely that smaller batch sizes lead to better model performance. In previous model training, the batch size was typically set to 8 [5, 10, 21] or 16 [14, 41]. Modern GPUs and TPUs are designed for highly parallel computation. Small batch size means there isn't enough data to fully saturate the computational units of these accelerators. This leads to idle cores and less effi-

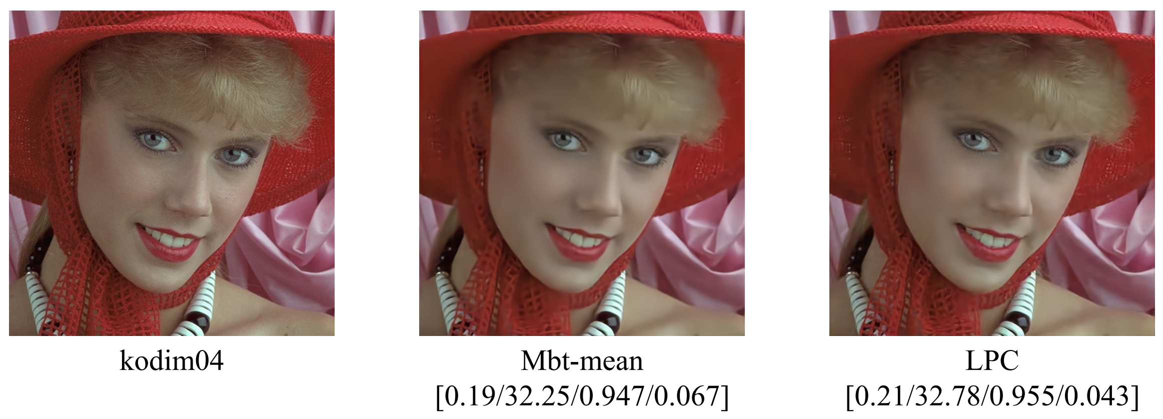


Figure 6. Visualization of the reconstructed images (*kodim04.png*) from Kodak dataset. The metrics are [bpp↓/PNSR↑/MS-SSIM↑/LPIPS↓]. *kodim04.png* shows LPC module achieves better detail reconstruction capabilities, for instance, in areas like teeth and hair.

cient use of the available processing power. Inverse Training trick effectively doubles the size of the training dataset. If the original batch size were maintained, the training time would consequently double. Therefore, we attempted to increase the original batch size from 8 to 16. Experimental results indicate that this led to a significant improvement in training speed with only a minor loss in performance, details are available at Tab. 5.

Dataset Size	Inverse Training	Eval Loss↓
300k	✓	0.7262 (-0.7%)
	✗	0.7313
150k	✓	0.7339 (-0.7%)
	✗	0.7387
75k	✓	0.7452 (-0.9%)
	✗	0.7522

Table 2. The Impact of Image Inverse Training on Different Training Dataset Sizes. We select Mbt-mean [27] as baseline and λ_1 is fixed at 0.0067.

Selection of Distribution Type. In Tab. 3, we further compare the latent compensation performance of the Distribution type. It is suggested that Compensation with Gaussian distribution achieves best performance. Consequently, we choose Gaussian distribution.

Type	Gaussian	Uniform	Laplacian	logistics
Loss	**0.7776**	0.8406	0.8574	0.8203

Table 3. Eval loss on four types of Compensation Distribution

Selection of Hyperparameter λ_2. In this section, we explore the impact of latent variable quantization loss weights λ_2 on model performance. The experimental results are shown in Tab. 4. Specifically, at $\lambda_2 = 0$, Eval Loss is 0.7313 and the whole training process degrades to origin optimization target. It then decreases to 0.7232 at $\lambda_2 = 0.25$, 0.7215 at $\lambda_2 = 0.5$, and hits its lowest point of 0.7184 at $\lambda_2 = 1$. However, as λ_2 increases further to 2 and 4, the Eval Loss begins to rise again, reaching 0.7326 and 0.7344 respectively. Based on the experiments above, we selected the hyperparameter λ_2 as 1.

Parameter	Bitrate(bpp)	PSNR(dB)	Eval Loss↓
$\lambda_2 = 0$	0.4075	31.70	0.7313
$\lambda_2 = 0.25$	0.4076	31.71	0.7232(-1.1%)
$\lambda_2 = 0.5$	0.4071	31.69	0.7215(-1.3%)
$\lambda_2 = 1$	0.4062	31.70	**0.7184(-1.7%)**
$\lambda_2 = 2$	0.4126	31.74	0.7326(+0.1%)
$\lambda_2 = 4$	0.4173	31.78	0.7344(+0.4%)

Table 4. Hyperparameter λ_2 selection. We select Mbt-mean[27] as baseline and λ_1 is fixed at 0.0067.

5. Conclusion

In this paper, we propose an easy-to-implement Latent compensation approach to boost VAE based image compression performance. Furthermore, We analyzed the training workflow of learned image compression models and consequently proposed two methods latent MSE regularization and inversed training. Extensive experiments demonstrate that our method succeeds in achieving better rate-distortion performance than baseline method. Moreover, we have comprehensively validated the effectiveness of each proposed component, highlighting the importance of learned compensation and appropriate hyperparameter selection. In the future, we will further explore the compatibility of our proposed module with SOTA methods.

References

[1] Workshop and challenge on learned image compression (clic). http://www.compression.cc, 2020. 6

[2] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc-Van Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. *Neural Information Processing Systems,Neural Information Processing Systems*, 2017. 1, 2

[3] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. 1

[4] Alex M. Andrew. *Brain Theory*, edited by günther palm and ad aertsen, springer, berlin, 1986 ix + 259 pp. (DM 96.)*Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. vol 1: Foundations. david e. rumelhart, james e. mcclelland and the PDF research group. MIT press, cambridge, mass., 1986, xx + 547 pp. (£27.50)*Parallel Distributed Processing: Explorations in the Microstructures of Cogition*. vol 2: Psychological and biological models. authorship, publisher, length and price all as for vol 1. *Robotica*, 5(4):344–345, 1987. 3

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 1, 3, 6, 7

[6] Johannes Ballé, Valero Laparra, and EeroP. Simoncelli. End-to-end optimized image compression. *International Conference on Learning Representations,International Conference on Learning Representations*, 2016. 2, 6

[7] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *CoRR*, abs/2011.03029, 2020. 6

[8] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6228–6237. Computer Vision Foundation / IEEE Computer Society, 2018. 6

[9] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 675–685. PMLR, 2019. 6

[10] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6, 7

[11] Haisheng Fu, Feng Liang, Jianping Lin, Bing Li, Mohammad Akbari, Jie Liang, Guohe Zhang, Dong Liu, Chengjie Tu, and Jingning Han. Learned image compression with gaussian-laplacian-logistic mixture model and concatenated residual modules. *IEEE Transactions on Image Processing*, 32:2063–2076, 2023. 3

[12] Chenjian Gao, Tongda Xu, Dailan He, Yan Wang, and Hongwei Qin. Flexible neural image compression via code editing. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 3

[13] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[14] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5708–5717. IEEE, 2022. 6, 7, 2

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017. 6

[16] Yueyu Hu, Wenhan Yang, and Jiaying Liu. Coarse-to-fine hyper-prior modeling for learned image compression. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 11013–11020, 2020. 3

[17] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy T. Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4385–4393. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[18] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 1, 3

[19] Eastman Kodak. Kodak lossless true color image suite (photocd pcd0992). http://r0k.us/graphics/kodak, 1993. 6

[20] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017. 6

[21] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. In *7th International Conference on*

Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. 7

[22] Mu Li, Wangmeng Zuo, Shuhang Gu, Jane You, and David Zhang. Learning content-weighted deep image compression. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3446–3461, 2021. 2

[23] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 14388–14397. IEEE, 2023. 6

[24] 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*, page 1247–1263, 2022. 2

[25] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[26] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, 2020. 1, 3

[27] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10794–10803, 2018. 3, 6, 7, 8, 1

[28] Yash Patel, Srikar Appalaraju, and R. Manmatha. Deep perceptual compression. *CoRR*, abs/1907.08310, 2019. 6

[29] Yash Patel, Srikar Appalaraju, and R. Manmatha. Human perceptual evaluations for image compression. *CoRR*, abs/1908.04187, 2019.

[30] Yash Patel, Srikar Appalaraju, and R. Manmatha. Saliency driven perceptual image compression. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 227–236, 2021. 6

[31] Yichen Qian, Zhiyu Tan, Xiuyu Sun, Ming Lin, Dongyang Li, Zhenhong Sun, Hao Li, and Rong Jin. Learning accurate entropy model with global reference for image compression. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3

[32] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, pages 1–5. IEEE, 2015. 1

[33] George Toderici, SeanM. O'Malley, SungJin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *International Conference on Learning Representations,International Conference on Learning Representations*, 2016. 2

[34] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5435–5443. IEEE Computer Society, 2017. 2

[35] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pages 1398–1402 Vol.2, 2003. 1

[36] Shao Xiang, Jing Xiao, and Mi Wang. A quantization loss compensation network for remote sensing image compression. In *Picture Coding Symposium, PCS 2024, Taichung, Taiwan, June 12-14, 2024*, pages 1–5. IEEE, 2024. 3

[37] Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. Enhanced invertible encoding for learned image compression. In *ACM MM*, pages 162–170, 2021. 1

[38] Yibo Yang, Robert Bamler, and Stephan Mandt. Improving inference for neural image compression. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3

[39] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. 6

[40] Yinhao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 2

[41] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image compression. In *CVPR*, 2022. 7