Correcting Quantization-Induced Gradient Mismatch in Neural Image Compression

Changhao Peng[1], **Yuqi Ye**[1], **Wei Gao**[1,2*]

[1]Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, China
[2] Peng Cheng Laboratory, Shenzhen, China

Abstract

In recent years, neural image compression methods have achieved impressive performance in image compression tasks, most of which are based on variational auto-encoder with hyper-prior and autoregressive Gaussian entropy model. We first demonstrate that the way these end-to-end approaches handle quantization during training leads to a mismatch between the gradients direction of entropy model parameters (i.e., mean and standard deviation) and the direction they should be optimized towards during inference, making neural network difficult to learn accurate estimates of entropy model parameters. To address this issue, we then propose a two-step improvement: in the first step, use straight-through estimator to align the forward propagation during training with inference, thereby correcting the gradients of standard deviation parameters; in the second step, utilize gradients transfer that we propose and MSE-guided gradients to manually compensate for the gradients of mean parameters lost due to straight-through estimator. Finally, we also propose to freeze the auto-encoder and hyper auto-encoder in pre-trained models provided by existing works, and fine-tune only the modules that predict the entropy model parameters, enabling efficient validation of proposed improvements. Experimental results show that our improvements bring appreciable performance gains to state-of-the-art neural image compression models in recent years. Meanwhile, our improvements require no modification to the structure of pre-trained models and only lightweight fine-tuning, which shows strong plug-and-play capability and practical utility.

Introduction

As a primary medium for human information acquisition, digital images have experienced explosive growth in the modern era. The widespread adoption of high-resolution images in mission-critical domains such as medical diagnostics (Sarvazyan 1998; Doi 2006), satellite remote sensing (Holmgren and Thuresson 1998; Lu et al. 2021a; Zeng, Guo, and Li 2022), and autonomous driving (Fujiyoshi, Hirakawa, and Yamashita 2019; Cui et al. 2021; Kaymak and Uçar 2019) poses severe challenges to storage and transmission infrastructures. Conventional codecs (e.g., JPEG (Wallace 1991), VVC (Bross et al. 2021)) employ handcrafted discrete cosine transforms (DCT) (Strang 1999) and entropy coding to achieve compression ratios (CR) of 10 to 20. However, these manually designed feature extractors face critical limitations: severe high-frequency detail loss and inadequate adaptation to semantic feature representations. Neural image compression frameworks address these challenges through end-to-end nonlinear transformation learning, while multi-scale neural architectures preserve more than 95% of perceptually salient features, establishing a new paradigm for bandwidth-efficient visual data delivery (Ballé et al. 2018).

Recent advancements in neural image compression (He et al. 2022; Liu, Sun, and Katto 2023; He et al. 2021; Jiang et al. 2023; Gao 2025), predominantly built upon variational auto-encoder (VAE) (Kingma, Welling et al. 2013) frameworks with hyper-prior and autoregressive entropy models proposed by Ballé et al (Ballé et al. 2018), have achieved remarkable rate-distortion performance. However, a critical yet overlooked challenge lies in the gradient mismatch induced by quantization during end-to-end training. Specifically, the gradients of entropy model parameters (e.g., mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$) are misaligned with their optimal update directions during inference, hindering accurate parameter estimation. This discrepancy arises because non-differentiable quantizations are approximated by adding uniform random noise to calculate likelihoods and utilizing straight-through estimators (STE) (Liu et al. 2022) to calculate reconstructed latents during training, which does not align the forward propagation of latent representations with inference. Consequently, $\boldsymbol{\mu}$ is driven toward perturbed values $(\boldsymbol{y} + \boldsymbol{\epsilon})$ rather than the original latent $\boldsymbol{y}$, while $\boldsymbol{\sigma}$ optimizes toward a suboptimal likelihood objective. Such mismatches degrade both compression efficiency and reconstruction fidelity.

To address this, we propose a two-step correction framework. First, we enforce consistency between training and inference by replacing noisy likelihood calculations with STE-quantized latents, rectifying the gradients of $\boldsymbol{\sigma}$. Second, we introduce gradient transfer and MSE-guided gradients to manually compensate for the lost gradients of $\boldsymbol{\mu}$ when $|\boldsymbol{y} - \boldsymbol{\mu}| < \frac{1}{2}$, ensuring $\hat{\boldsymbol{y}}$ aligns precisely with $\boldsymbol{y}$. Crucially, our method operates as a plug-and-play enhancement as it does not need to modify the model architecture: by freezing auto-encoders and hyper-auto-encoders of pre-trained models provided by current state-of-the-art works, we efficiently

*Corresponding author.

fine-tune only the entropy parameter modules, minimizing computational overhead.

Therefore, our contributions can be summarized as follows:

- **Uncovering gradient mismatch in quantization approximation:** We identify and formalize a critical flaw in existing end-to-end neural image compression frameworks: the use of uniform noise and straight-through estimators during training introduces a misalignment between gradient updates for entropy model parameters and their optimal inference-time directions, which could degrade rate-distortion performance.
- **Two-phase correction framework:** To resolve this mismatch, we propose a two-phase correction framework. Firstly, align forward propagation during training with inference by replacing noisy likelihood approximations with STE-quantized latents, rectifying gradient directions of $\boldsymbol{\sigma}$. Secondly, introduce gradient transfer and MSE-guided gradient compensation to manually restore lost gradients for $\boldsymbol{\mu}$ when $|\boldsymbol{y} - \boldsymbol{\mu}| < \frac{1}{2}$, ensuring precise alignment between $\boldsymbol{\mu}$ and $\boldsymbol{y}$ for optimal reconstruction.
- **Plug-and-play capability via partial fine-tuning:** We demonstrate that freezing pre-trained auto-encoders and hyper-auto-encoders while fine-tuning only entropy parameter modules enables rapid validation of our improvements, which shows strong plug-and-play capability as our improvements require no modification to the structure of pre-trained models. Experimental results show that with minimal fine-tuning, our approach achieves notable rate-distortion performance improvements over the current state-of-the-art neural image compression models.

Related Work

As recent advances (Minnen and Singh 2020; Minnen, Ballé, and Toderici 2018; He et al. 2021, 2022; Jiang et al. 2023; Liu, Sun, and Katto 2023; Han et al. 2024; Xie, Cheng, and Chen 2021; Yang et al. 2021; Lin et al. 2025; Wang et al. 2025) in neural image compression have predominantly centered on VAE-based frameworks with hyper-prior and entropy models, these models achieve compression by learning latent representations through an auto-encoder and optimizing rate-distortion trade-offs via entropy-constrained training. Therefore, current improvements largely focus on two directions: enhancing auto-encoder architectures and refining entropy model estimation. Below, we synthesize recent progress in these areas.

Improving Auto-Encoder Architectures

In neural image compression, auto-encoders serve to reduce the dimensionality of images and learn compact latent representations, which typically comprise two components: one alters spatial dimensions (i.e., downsampling and upsampling), and another transforms features for more efficient representation while maintaining dimensions. Convolutions and transposed convolutions are commonly employed for spatial dimension changes, while existing works mainly focus on various nonlinear transforms for effective latent representation.

Pioneering work by Ballé et al. (Ballé et al. 2018) introduces the use of generalized divisive normalization (GDN) (Ballé, Laparra, and Simoncelli 2015) to capture effective latent representations, which has been subsequently replaced or augmented by convolution neural networks (CNN) (LeCun et al. 1989) and more complex nonlinear transforms in follow-up studies (Chen et al. 2021; Ma et al. 2020; Gao et al. 2021; Xie, Gao, and Zheng 2022). Recently, vision transformer (Dosovitskiy et al. 2020) has achieved promising results in numerous visual tasks, leading to the incorporation of vision transformer and its variants into auto-encoder to obtain better latent representations of images (Lu et al. 2021b; Zhu, Yang, and Cohen 2022). However, this also increases the time required for encoding/decoding and the difficulty of training. Liu et al. (Liu, Sun, and Katto 2023) propose extracting local features using CNNs and global features using Swin Transformers (Liu et al. 2021), leveraging the strengths of both CNNs and Swin Transformers to improve rate-distortion performance.

Improving Entropy Model Estimation

Neural image compression models typically use arithmetic coding (Witten, Neal, and Cleary 1987) to convert latents into binary bitstreams. The parameters required for the probability entropy model in arithmetic coding are estimated by neural networks. In the work of Ballé et al. (Ballé et al. 2018), only hyper-prior is used as side information to predict the entropy model parameters. While this approach enables fast encoding and decoding, it limits rate-distortion performance. Minnen et al. (Minnen, Ballé, and Toderici 2018) propose using masked convolution in an autoregressive manner to estimate entropy model parameters. Although this method exploits partial contextual information, it significantly reduces the speed of encoding and decoding. He et al. (He et al. 2021) propose a checkerboard model to evenly divide the latents into two parts, using the anchor part as contextual information to predict the entropy model parameters of the non-anchor part, thereby efficiently utilizing already decoded information. He et al. (He et al. 2022) propose an uneven channel-wise partitioning strategy, conducting multi-step encoding and decoding across channels. Subsequent studies have largely adopted multi-step channel-wise decoding combined with two-step intra-channel decoding using the checkerboard pattern. The main improvements (Kong, Sun, and Xue 2024; Guo et al. 2022; Zhang et al. 2025) in these works involve employing more efficient modules (e.g., RWKV (Peng et al. 2023), Mamba (Gu and Dao 2023)) for entropy parameter estimation.

Preliminary: Variational Compression Model with Hyper-Prior

In variational compression model with hyper-prior, image $\boldsymbol{x}$ is firstly encoded by auto-encoder g_a to get the latent representation $\boldsymbol{y}$, and further processed by hyper auto-encoder h_a to extract the hyper latent $\boldsymbol{z}$:

$$\boldsymbol{y} = g_a(\boldsymbol{x}), \boldsymbol{z} = h_a(\boldsymbol{y}). \tag{1}$$

By estimating the probability models of $\boldsymbol{y}$ and $\boldsymbol{z}$, we can encode $\boldsymbol{y}$ and $\boldsymbol{z}$ into binary strings using arithmetic cod-

ing. Previous works predominantly utilize Gaussian entropy models (or mixture Gaussian entropy models) to perform probability estimation for $\boldsymbol{y}$ and $\boldsymbol{z}$:

$$\begin{aligned} p_{\hat{\boldsymbol{z}}|\boldsymbol{\psi}}(\hat{\boldsymbol{z}}|\boldsymbol{\psi}) &= \prod_i \left[p_{z_i|\psi_i}(\psi_i) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right] (\hat{z}_i), \\ p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) &= \prod_i \left[\mathcal{N}(\mu_i, \sigma_i^2) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right] (\hat{y}_i), \end{aligned} \tag{2}$$

where $\boldsymbol{\psi}$ is a non-parametric fully factorized density model (Ballé et al. 2018), entropy model parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are estimated based on quantized and decoded $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$. Finally, $\hat{\boldsymbol{y}}$ is decoded to obtain the reconstructed image:

$$\hat{\boldsymbol{x}} = g_s(\hat{\boldsymbol{y}}). \tag{3}$$

During end-to-end training, the quantization operation is non-differentiable. Previous works (He et al. 2021; Minnen, Ballé, and Toderici 2018; He et al. 2022; Lu et al. 2025; Li et al. 2025; Han et al. 2024) address this by adding uniform noise $\epsilon \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ when computing the likelihood during training:

$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) = \prod_i \left[\mathcal{N}(\mu_i, \sigma_i^2) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right] (y_i + \epsilon_i), \tag{4}$$

and utilizing straight-through estimator to compute the decoded $\hat{\boldsymbol{y}}$:

$$\begin{aligned} \hat{\boldsymbol{y}} &= ste(\boldsymbol{y} - \boldsymbol{\mu}) + \boldsymbol{\mu} \\ &= round(\boldsymbol{y} - \boldsymbol{\mu}) + (\boldsymbol{y} - \boldsymbol{\mu}) - (\boldsymbol{y} - \boldsymbol{\mu}).detach() + \boldsymbol{\mu} \\ &= round(\boldsymbol{y} - \boldsymbol{\mu}) + \boldsymbol{y} - (\boldsymbol{y} - \boldsymbol{\mu}).detach(). \end{aligned} \tag{5}$$

Compared to hyper latent $\boldsymbol{z}$, latent $\boldsymbol{y}$ constitutes the majority of the bitstream (usually more than 95%). Hence, our focus is on the impact of quantization operations on $\boldsymbol{y}$.

Inappropriate Gradient

Previous works typically use rate-distortion cost (Berger 2003) as the loss function for end-to-end training:

$$\begin{aligned} \mathcal{L} =& R + \lambda \cdot D \\ =& E_{\boldsymbol{x} \sim p_{\boldsymbol{x}}} \left[-\log_2 p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) - \log_2 p_{\hat{\boldsymbol{z}}}(\hat{\boldsymbol{z}}) \right] \\ &+ \lambda \cdot E_{\boldsymbol{x} \sim p_{\boldsymbol{x}}}[d(\boldsymbol{x}, \hat{\boldsymbol{x}})]. \end{aligned} \tag{6}$$

Note that $round(\boldsymbol{y} - \boldsymbol{\mu})$ and $(\boldsymbol{y} - \boldsymbol{\mu}).detach()$ are non-differentiable in Equation 5, we have:

$$\nabla_{\boldsymbol{\mu}} \hat{\boldsymbol{y}} = \mathbf{0}. \tag{7}$$

Thus, the gradient of the distortion term in the loss function with respect to $\boldsymbol{\mu}$ is:

$$\begin{aligned} \nabla_{\boldsymbol{\mu}} E_{\boldsymbol{x} \sim p_{\boldsymbol{x}}}[d(\boldsymbol{x}, \hat{\boldsymbol{x}})] &= \nabla_{\boldsymbol{\mu}} E_{\boldsymbol{x} \sim p_{\boldsymbol{x}}}[d(\boldsymbol{x}, g_s(\hat{\boldsymbol{y}}))] \\ &= \nabla_{\hat{\boldsymbol{y}}} E_{\boldsymbol{x} \sim p_{\boldsymbol{x}}}[d(\boldsymbol{x}, g_s(\hat{\boldsymbol{y}}))] \cdot \nabla_{\boldsymbol{\mu}} \hat{\boldsymbol{y}} \\ &= \mathbf{0}, \end{aligned} \tag{8}$$

which implies that the distortion term in the loss function is incapable of optimizing the parameter estimation of the entropy model (since $\hat{\boldsymbol{y}}$ is independent of $\boldsymbol{\sigma}$, $\nabla_{\boldsymbol{\sigma}} E_{\boldsymbol{x} \sim p_{\boldsymbol{x}}}[d(\boldsymbol{x}, \hat{\boldsymbol{x}})] = \mathbf{0}$ also holds true). Therefore, the gradient of the loss function with respect to $\boldsymbol{\mu}$ depends on the rate term:

$$\begin{aligned} \nabla_{\boldsymbol{\mu}} \mathcal{L} &= \nabla_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{x}}} \left[-\log_2 p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) - \log_2 p_{\hat{\boldsymbol{z}}}(\hat{\boldsymbol{z}}) \right] \\ &= \nabla_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{x}}} \left[-\log_2 p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) \right] \\ &= -\sum_i \nabla_{\mu_i} \log_2 \left[\mathcal{N}(\mu_i, \sigma_i^2) * \mathcal{U}\left(-\tfrac{1}{2}, \tfrac{1}{2}\right) \right] (y_i + \varepsilon_i) \\ &= -\sum_i \nabla_{\mu_i} f(\mu_i, y_i + \varepsilon_i, \sigma_i). \end{aligned} \tag{9}$$

where

$$f(\mu, k, \sigma) = -\log_2 \left[\int_{-\frac{1}{2}}^{\frac{1}{2}} \exp\left(-\frac{(k + t - \mu)^2}{2\sigma^2} \right) dt \right]. \tag{10}$$

To further investigate the gradient direction of $\boldsymbol{\mu}$, we compute the partial derivative of Equation 9 with respect to $\boldsymbol{\mu}$:

$$\begin{aligned} \frac{\partial f}{\partial \mu} =& \frac{-1}{\sigma^2 \ln(2) \int_{-\frac{1}{2}}^{\frac{1}{2}} \exp\left(-\frac{(k+t-\mu)^2}{2\sigma^2} \right) dt} \\ & \cdot \int_{-\frac{1}{2}}^{\frac{1}{2}} (k + t - \mu) \exp\left(-\frac{(k + t - \mu)^2}{2\sigma^2} \right) dt. \end{aligned} \tag{11}$$

When $\mu = y + \epsilon$, we have:

$$\begin{aligned} \frac{\partial f}{\partial \mu} &= \frac{-1/\ln(2)}{\sigma^2 \int_{-\frac{1}{2}}^{\frac{1}{2}} \exp\left(-\frac{t^2}{2\sigma^2}\right) dt} \int_{-\frac{1}{2}}^{\frac{1}{2}} t \exp\left(-\frac{t^2}{2\sigma^2} \right) dt \\ &= 0. \end{aligned} \tag{12}$$

Since f is a convex function with respect to μ (Roberts and Varberg 1974), it takes the minimum value when $\mu = y + \epsilon$. Therefore, the gradient of the loss function with respect to $\boldsymbol{\mu}$ drives $\boldsymbol{\mu}$ towards $\boldsymbol{y} + \boldsymbol{\epsilon}$. However, in encoding and decoding processes, it is preferable for $\boldsymbol{\mu}$ to be closer to $\boldsymbol{y}$. Consequently, when the relative magnitudes of $\boldsymbol{y}$ and $\boldsymbol{y} + \boldsymbol{\epsilon}$ with respect to $\boldsymbol{\mu}$ differ (for example, $\boldsymbol{y} < \boldsymbol{\mu} < \boldsymbol{y} + \boldsymbol{\epsilon}$), $\boldsymbol{\mu}$ will be updated in an incorrect direction, which means that the gradient of the loss function with respect to $\boldsymbol{\mu}$ is inappropriate under such circumstances.

An Intuitive Correction

In Section , we demonstrated that the introduction of uniform random noise during training will affect the accuracy of the gradient of $\boldsymbol{\mu}$. An intuitive correction is to eliminate uniform random noise when calculating the likelihood:

$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) = \prod_i \left[\mathcal{N}(\mu_i, \sigma_i^2) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right] (y_i). \tag{13}$$

In this way, the gradient of the loss function with respect to $\boldsymbol{\mu}$ consistently points towards $\boldsymbol{y}$. However, we will subsequently

illustrate that this correction introduces a new problem: it leads to inaccurate estimation of $\boldsymbol{\sigma}$.

Similar to Equation 9, the gradient of the loss function with respect to $\boldsymbol{\sigma}$ can be expressed as:

$$\nabla_{\boldsymbol{\sigma}}\mathcal{L} = \nabla_{\boldsymbol{\sigma}} \sum_i \log_2 \sqrt{2\pi\sigma_i^2} - \log_2 f(\mu_i, y_i, \sigma_i). \quad (14)$$

Thus, $\boldsymbol{\sigma}$ moves towards the optimal $\boldsymbol{\sigma}^{\boldsymbol{opt\_train}}$ during training:

$$\boldsymbol{\sigma}^{\boldsymbol{opt\_train}} = \arg\min_{\boldsymbol{\sigma}} \sum_i \log_2 \sqrt{2\pi\sigma_i^2} - \log_2 f(\mu_i, y_i, \sigma_i). \quad (15)$$

However, during encoding and decoding processes, it is $round(\boldsymbol{y} - \boldsymbol{\mu})$ that participates in the likelihood calculation, not $\boldsymbol{y} - \boldsymbol{\mu}$. Therefore, the optimal $\boldsymbol{\sigma}^{\boldsymbol{opt}}$ is actually:

$$\begin{aligned}\boldsymbol{\sigma}^{\boldsymbol{opt}} = &\arg\min_{\boldsymbol{\sigma}} \sum_i \log_2 \sqrt{2\pi\sigma_i^2} \\ &- \sum_i \log_2 f(0, round(y_i - \mu_i), \sigma_i).\end{aligned} \quad (16)$$

Figure 1 illustrates the differences between σ^{opt} and σ^{opt_train} as $y - \mu$ varies. It can be observed that removing the uniform random noise when calculating the likelihood during training leads to the optimization of $\boldsymbol{\sigma}$ in an incorrect direction.

Therefore, we believe that to fully address the aforementioned issue of inappropriate gradient during training, it is necessary to ensure consistency between the reconstruction of the latent $\boldsymbol{y}$ and the calculation of likelihood $p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}})$ during both training and encoding/decoding processes. Consequently, we propose a two-step improvement method: First, it is necessary to ensure consistency between computations in forward propagation during training and inference. Then, for the gradients lost due to quantization, we need to manually set them.

Keep Training and Inference Consistent

There are two points where quantization of latent $\boldsymbol{y}$ is involved during inference: the reconstruction of $\boldsymbol{y}$ and the calculation of its likelihood. Equation 5 ensures that the reconstruction of $\boldsymbol{y}$ during inference and training is consistent, so we only need to modify the likelihood calculation during training:

$$p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) = \left[\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right)\right](ste(\boldsymbol{y} - \boldsymbol{\mu}) + \boldsymbol{\mu}). \quad (17)$$

Equation 17 utilizes reconstructed $\hat{\boldsymbol{y}} = ste(\boldsymbol{y} - \boldsymbol{\mu}) + \boldsymbol{\mu}$ to calculate the likelihood, thereby maintaining consistency between training and inference. However, this approach gives rise to a new issue: when $|y - \mu| < \frac{1}{2}$, we have:

$$\frac{\partial p_{\hat{y}|\hat{\boldsymbol{z}}}(\hat{y}|\hat{\boldsymbol{z}})}{\partial \mu} = \frac{\partial \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{t^2}{2\sigma^2}\right) dt}{\partial \mu} = 0, \quad (18)$$

and in conjunction with Equation 7, it can be observed that when $|y - \mu| < \frac{1}{2}$, the gradient of the loss function with respect to μ is:

$$\nabla_{\mu}\mathcal{L} = 0. \quad (19)$$

Therefore, the network will no longer optimize the estimation of μ if $|y - \mu| < \frac{1}{2}$ during training. However, while μ can ensure the optimal likelihood in this case, it cannot guarantee the optimal reconstructed latent:

$$\hat{y} = round(y - \mu) + \mu = \mu. \quad (20)$$

The discrepancy between the reconstructed $\hat{y} = \mu$ and y cannot be further optimized at this time, and we propose to manually set the gradients to address this issue.

Manually Set Gradients When $|y - \mu| < \frac{1}{2}$

Our core idea is to manually set the gradients with respect to μ when $|y - \mu| < \frac{1}{2}$, so that μ continues to approach y, leading to more accurate reconstructed $\hat{y}$. To facilitate optimization, it is preferable to keep $\boldsymbol{y}$ fixed during this process. Therefore, we propose to freeze the auto-encoder (g_a, g_s) and hyper auto-encoder (h_a, h_s) in the pre-trained model provided by previous work, and fine-tune only the remaining components. This approach not only keeps $\boldsymbol{y}$ fixed for stable optimization but also accelerates training.

Upon experimentation, we have found that the gradient for μ can be set in the following two ways when $|y - \mu| < \frac{1}{2}$.

MSE-Guided Gradients

A straightforward idea is: since we want μ to continue approaching y when $|y - \mu| < \frac{1}{2}$, we can directly add MSE loss between $\boldsymbol{y}$ and $\boldsymbol{\mu}$ to the loss function:

$$\begin{aligned}\mathcal{L} = &E_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}\left[-\log_2 p_{\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{z}}) - \log_2 p_{\hat{\boldsymbol{z}}}(\hat{\boldsymbol{z}})\right] \\ &+ \lambda_1 \cdot E_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[d(\boldsymbol{x}, \hat{\boldsymbol{x}})] + \lambda_2 \cdot \text{MSE}(\boldsymbol{y}, \overline{\boldsymbol{\mu}}).\end{aligned} \quad (21)$$

To avoid affecting the gradients when $|y - \mu| \geq \frac{1}{2}$, we set:

$$\overline{\mu}_i = \begin{cases} \mu & \text{if } round(y - \mu) = 0 \\ y & \text{if } round(y - \mu) \neq 0 \end{cases}. \quad (22)$$

Thus, additional gradients will only be generated when $|y - \mu| < \frac{1}{2}$, thereby causing μ to move towards y.

Gradients Transfer

It can be seen that $\hat{y} = \mu$ when $|y - \mu| < \frac{1}{2}$ from Equation 20. At this point, the gradient of the loss function with respect to $\hat{y}$ can be regarded as the direction in which μ should move. However, as indicated by Equation 5, $\hat{\boldsymbol{y}}$ has gradients only with respect to $\boldsymbol{y}$ ($\frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{y}} = \boldsymbol{1}$), but not with respect to $\boldsymbol{\mu}$. Therefore, we can transfer the gradient of the loss function with respect to $\boldsymbol{y}$ to $\boldsymbol{\mu}$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_{|\boldsymbol{y}-\boldsymbol{\mu}|<\frac{1}{2}}} &:= k \cdot \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{y}}_{|\boldsymbol{y}-\boldsymbol{\mu}|<\frac{1}{2}}} \cdot \frac{\partial \hat{\boldsymbol{y}}_{|\boldsymbol{y}-\boldsymbol{\mu}|<\frac{1}{2}}}{\partial \boldsymbol{y}} \\ &= k \cdot \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{y}}_{|\boldsymbol{y}-\boldsymbol{\mu}|<\frac{1}{2}}},\end{aligned} \quad (23)$$

where k is a hyperparameter used to ensure the stability of training. Algorithm 1 presents the Python implementation for gradients transfer we propose, which can be implemented within 20 lines of Python code.

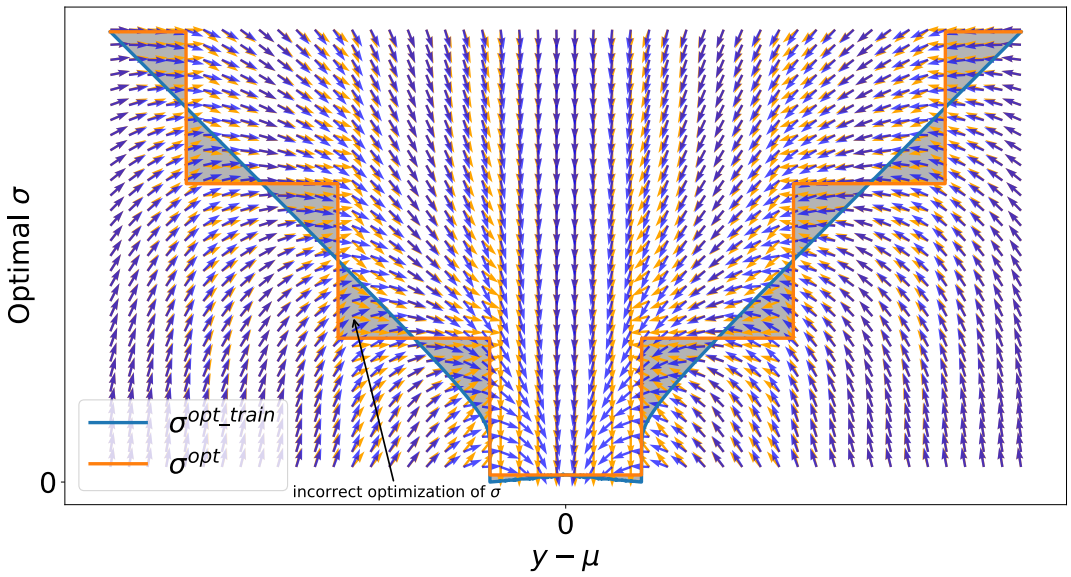

Figure 1: The difference between the optimal σ^{opt_train} during training and the optimal σ^{opt} during inference when $y - \mu$ varies. The arrows with corresponding colors represent the gradient directions during the training/inference of $(y - \mu, \sigma)$. σ will be optimized in the wrong direction when $(y - \mu, \sigma)$ is within the gray area of the graph while $y - \mu$ will always be optimized towards 0 during training.

Experimental Results

Fine-Tuning on Pre-Trained Models

On the one hand, current mainstream deep image compression models handle quantization during training as described in Chapter , on the other hand, our proposed method does not alter the model structure. Therefore, to validate the effectiveness of proposed method in a simple and efficient manner, we argue that fine-tuning the pre-trained models provided by recent state-of-the-art works (He et al. 2021, 2022; Liu, Sun, and Katto 2023; Jiang et al. 2025; Li et al. 2023; Han et al. 2024; Lu et al. 2025) is a viable approach.

As described in Chapter , we manually set gradients to keep $\boldsymbol{\mu}$ moving closer to $\boldsymbol{y}$ when $|y - \mu| < \frac{1}{2}$, with $\boldsymbol{y}$ ideally fixed to allow the network to better optimize the estimation of $\boldsymbol{\mu}$. Therefore, we freeze the auto-encoder (g_a, g_s) and hyper auto-encoder (h_a, h_s) in the pre-trained models during fine-tuning and train the remaining part that predicts parameters of entropy model, which not only makes it easier to train but also accelerates training.

Implementation Details

We fine-tune the pre-trained models with our improvement using the dataset proposed by MLIC++ (Jiang et al. 2025), which contains 10^5 images. Images are randomly cropped to a size of 256×256 and the batch size is set as 16 during training. We simply set $\lambda_2 = 1$ in Equation 21. The training is performed on Intel Core i9-10900K CPU @ 3.60GHz and NVIDIA GeForce RTX 3090 GPU for 50 epochs, with learning rates of 10^{-5} for the first 40 epochs and 10^{-6} for last 10 epochs. To eliminate irrelevant factors, we also train the pre-trained models without our improvement on the same dataset under identical settings and compared these results with those obtained from the improved models.

Rate-Distortion Performance

Figure 2 presents the rate-distortion curves demonstrating the performance of different methods, where Peak Signal-to-Noise Ratio (PSNR) represents the distortion level of decoded images, and the bitrate is represented by bits per pixel (bpp). It can be seen from the figure that our improvements bring notable performance gains to all base models.

Table 1 illustrates the effects of our improvements on state-of-the-art neural image compression models from recent years, where BD-BR (Bitrate-Distortion Benefit Ratio) is calculated with VVC (VTM-12.1 Intra) (Bross et al. 2021) as the anchor. We exclude the influence of irrelevant factors such as data and training settings through the results of "Fine-Tune" column in Table 1. The results presented in Table 1 indicate that both MSE-guided gradients and gradients transfer can enhance base models' encoding performance to a certain extent, while combining MSE-guided gradients and gradients transfer further improves performance. Meanwhile, for base models with better performance, such as DCAE (Lu et al. 2025), the performance gain brought by our improvements is relatively small. We believe this is because the powerful auto-encoder and entropy model of these models mitigate the impact of quantization-induced gradient mismatch during training.

Since our method only improves the training of base models without altering their architecture, the encoding and decoding time during inference will remain unchanged.

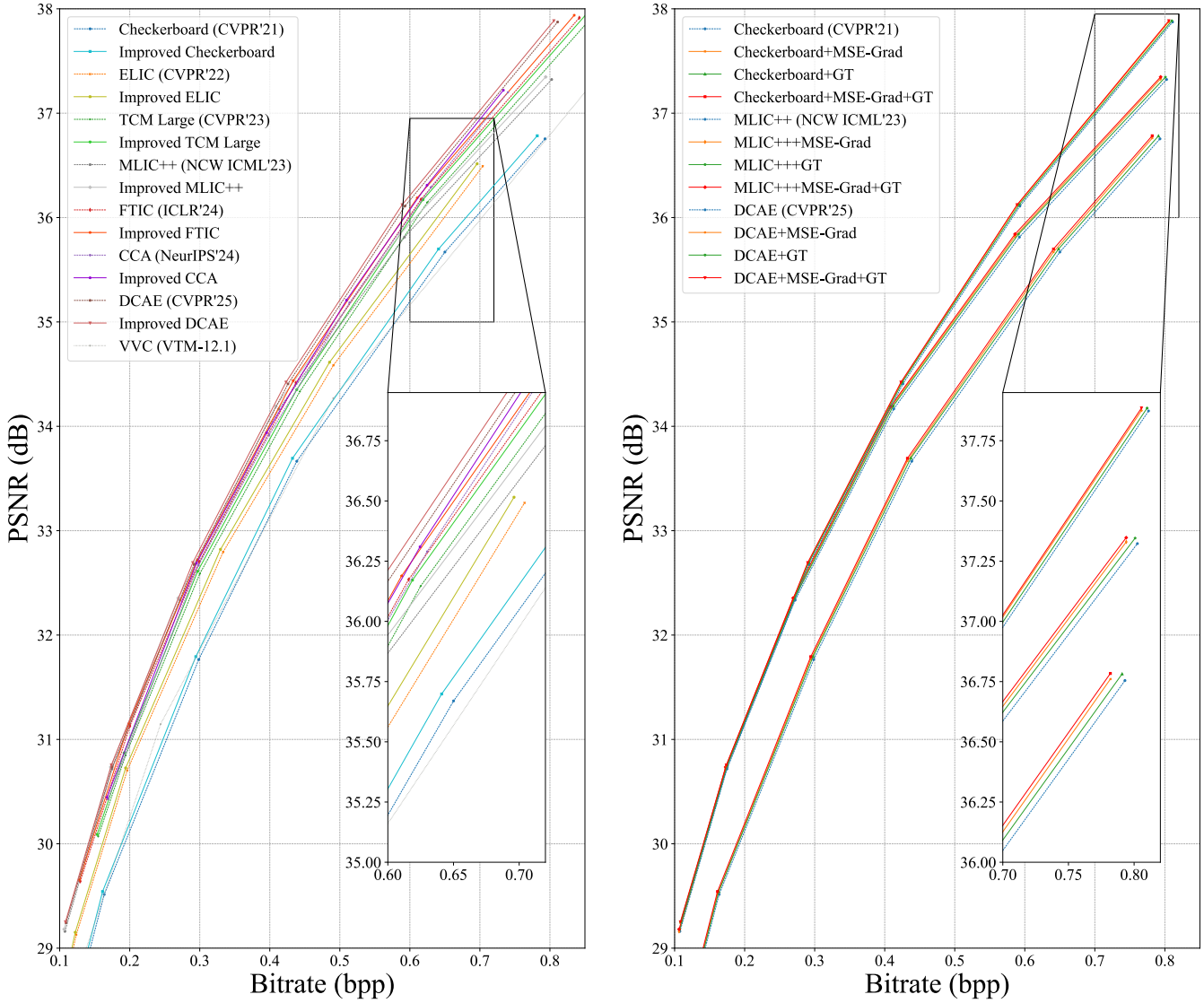

Figure 2: (Left) Rate-distortion curves of recent state-of-the-art models with and without our improvement on Kodak dataset. (Right) Ablation study for MSE-Grad (MSE-guided gradients) and GT (gradients transfer) based on Checkerboard, MLIC++ and DCAE.

Model	**Improvement**				
	None	**Fine-Tuning**	**MSE-Grad**	**GT**	**MSE-Grad+GT**
Checkerboard (CVPR'21) (He et al. 2021)	3.29	3.29	1.79	2.32	**1.20**
ELIC (CVPR'22) (He et al. 2022)	-8.00	-7.99	-9.14	-8.70	**-9.55**
TCM (CVPR'23) (Liu, Sun, and Katto 2023)	-12.03	-12.03	-12.97	-12.59	**-13.31**
MLIC++ (NCW ICML'23) (Jiang et al. 2025)	-15.26	-15.26	-16.22	-15.86	**-16.56**
FTIC (ICLR'24) (Li et al. 2023)	-14.95	-14.93	-15.79	-15.42	**-16.02**
CCA(NeurIPS'24) (Han et al. 2024)	-13.77	-13.76	-14.57	-14.26	**-14.86**
DCAE (CVPR'25) (Lu et al. 2025)	-17.36	-17.34	-17.98	-17.73	**-18.16**

Table 1: BD-BR↓ (%) of recent state-of-the-art models. "Fine-Tuning" denotes training under the same settings on the training dataset without applying any proposed improvement to the models.

Listing 1: Gradients transfer.

```
# Input: y (latent);  mu (expected value
    of entropy model)
# Output: y_hat (reconstructed latent)

import torch

class MyQuantizer(torch.autograd.
    Function):
    @staticmethod
    def forward(ctx, y, mu):
        ctx.save_for_backward(y, mu)
        return torch.round(y - mu) + mu

    @staticmethod
    def backward(ctx, grad_output):
        k = 1e-2   #Set hyperparameter k
            to ensure the stability of
            training.
        y, mu = ctx.saved_tensors
        mask = (torch.round(y - mu) ==
            0) #Gradients transfer is
            performed
        grad_mu = torch.where(mask, k *
            grad_output, 0.) #Transfer
            the gradients with respect to
            y to mu.
        grad_y  = grad_output
        return grad_y, grad_mu

def my_quantizer(y, mu):
    return MyQuantizer.apply(y, mu)

y_hat = my_quantizer(y, mu)
```

Ablation Study

By aligning the corresponding bitrate points in Figure 2, it can be observed that the gain of MSE-guided gradients is primarily due to lower bpp, while the gain of gradients transfer is mainly attributed to lower PSNR, which means less distortion of the reconstructed images.

On the one hand, MSE-guided gradients enables the $\boldsymbol{\mu}$ to continue approaching the $\boldsymbol{y}$ even when $|\boldsymbol{y}-\boldsymbol{\mu}|<\frac{1}{2}$. Furthermore, since the computation processes for forward propagation and inference are aligned, $\boldsymbol{\sigma}$ is simultaneously optimized toward the optimal $\boldsymbol{\sigma^{opt}}$ during inference as $\boldsymbol{\mu}$ moves closer to $\boldsymbol{y}$. Consequently, the improved entropy model parameters reduce the bitstream. Additionally, because a more accurate $\boldsymbol{\mu}$ yields a more precise reconstructed $\hat{\boldsymbol{y}}$, MSE-guided gradients can also improve PSNR to some extent. We calculate the MSE between the entropy model parameters of ELIC before and after optimization utilizing MSE-guided gradients and the optimal entropy model parameters. The results are shown in Figure 3, which intuitively demonstrates that MSE-guided gradients allow the model to estimate the entropy model parameters more accurately. Results about MSE-guided gradients in Table 2 demonstrate that the original model yields correct gradients for μ when $round(y-\mu)\neq 0$, and introducing additional manually-set gradients under this condition degrades model performance.

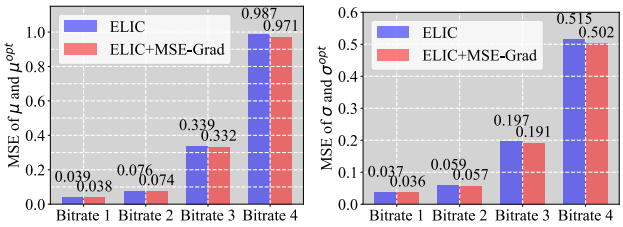

Figure 3: MSE between the optimal entropy parameters during inference and those predicted by ELIC with and without MSE-guided gradients. To highlight the differences, we test ELIC at its four highest bitrate points, with bitrates increasing sequentially from Bitrate 1 to Bitrate 4.

Improvement	Setting		Result
	condition	k	
None	/	/	3.29
Fine-Tuning	/	/	3.29
MSE-Grad	w/o	/	1.84
MSE-Grad	w/	/	1.79
GT	/	0.1	NA
GT	/	0.01	2.32
GT	/	0.001	2.85

Table 2: BD-BR↓ (%) of Checkerboard under different improvement configurations. "Condition" denotes whether MSE-guided gradients is applied exclusively when $round(y-\mu)=0$.

On the other hand, gradients transfer leverages the property where $\hat{\boldsymbol{y}}=ste(\boldsymbol{y}-\boldsymbol{\mu})+\boldsymbol{\mu}=\boldsymbol{\mu}$ when $|\boldsymbol{y}-\boldsymbol{\mu}|<\frac{1}{2}$, transferring the gradients of $\hat{\boldsymbol{y}}$ with respect to $\boldsymbol{y}$ to $\boldsymbol{\mu}$. Since $\hat{\boldsymbol{y}}$ is subsequently used for decoding, this approach enables $\boldsymbol{\mu}$ to move towards lower distortion when $|\boldsymbol{y}-\boldsymbol{\mu}|<\frac{1}{2}$. Therefore, gradients transfer can improve PSNR, but it does not significantly reduce bpp. As shown in Table 2,we discover that that $k>0.01$ may cause training instability(NaN), while smaller k degrades performance.

Conclusion

We first point out that current end-to-end neural image compression methods handle quantization inappropriately during training, making it difficult to learn accurate entropy parameters. This deficiency arises from the discrepancy between the differentiable proxy and actual discrete quantization. Subsequently, we attempt to remove random uniform noise from training, but find that this also leads to gradient directions misaligned with those required for optimal inference. To address this issue, we propose aligning the forward propagation during training with inference, and then manually specifying the gradients with respect to the entropy parameters during backpropagation. Our approach yields appreciable improvements in compression performance without modifying the model architecture or increasing inference time, and requires only minimal fine-tuning of a pre-trained model, highlighting its plug-and-play capability.

Acknowledgments

This work was supported by The Major Key Project of PCL (PCL2024A02), Natural Science Foundation of China (62271013, 62031013), Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (2024B1212010006), Guangdong Province Pearl River Talent Program (2021QN020708), Guangdong Basic and Applied Basic Research Foundation (2024A1515010155), Shenzhen Science and Technology Program (JCYJ20240813160202004, JCYJ20230807120808017, SYSPG20241211173440004), Shenzhen Fundamental Research Program (GXWD20201231165807007-20200806163656003).

References

Ballé, J.; Laparra, V.; and Simoncelli, E. P. 2015. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*.

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

Berger, T. 2003. Rate-distortion theory. *Wiley Encyclopedia of Telecommunications*.

Bross, B.; Wang, Y.-K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G. J.; and Ohm, J.-R. 2021. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10): 3736–3764.

Chen, T.; Liu, H.; Ma, Z.; Shen, Q.; Cao, X.; and Wang, Y. 2021. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 30: 3179–3191.

Cui, Y.; Chen, R.; Chu, W.; Chen, L.; Tian, D.; Li, Y.; and Cao, D. 2021. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2): 722–739.

Doi, K. 2006. Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology. *Physics in Medicine & Biology*, 51(13): R5.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fujiyoshi, H.; Hirakawa, T.; and Yamashita, T. 2019. Deep learning-based image recognition for autonomous driving. *IATSS research*, 43(4): 244–252.

Gao, G.; You, P.; Pan, R.; Han, S.; Zhang, Y.; Dai, Y.; and Lee, H. 2021. Neural image compression via attentional multi-scale back projection and frequency decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14677–14686.

Gao, W. 2025. *AI-based Image and Video Coding: Methods, Standards, and Applications*. Springer Nature.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Guo, Z.; Zhang, Z.; Feng, R.; and Chen, Z. 2022. Causal Contextual Prediction for Learned Image Compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(4): 2329–2341.

Han, M.; Jiang, S.; Li, S.; Deng, X.; Xu, M.; Zhu, C.; and Gu, S. 2024. Causal Context Adjustment Loss for Learned Image Compression. *arXiv preprint arXiv:2410.04847*.

He, D.; Yang, Z.; Peng, W.; Ma, R.; Qin, H.; and Wang, Y. 2022. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5718–5727.

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

Holmgren, P.; and Thuresson, T. 1998. Satellite remote sensing for forestry planning—a review. *Scandinavian Journal of Forest Research*, 13(1-4): 90–110.

Jiang, W.; Yang, J.; Zhai, Y.; Gao, F.; and Wang, R. 2025. MLIC++: Linear Complexity Multi-Reference Entropy Modeling for Learned Image Compression. *ACM Transactions on Multimedia Computing, Communications, and Applications*.

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

Kaymak, Ç.; and Uçar, A. 2019. A brief survey and an application of semantic image segmentation for autonomous driving. *Handbook of deep learning applications*, 161–200.

Kingma, D. P.; Welling, M.; et al. 2013. Auto-encoding variational bayes.

Kong, W.; Sun, M.; and Xue, H. 2024. FANs: fully attentional networks for image compression. *Multimedia Tools and Applications*, 83(17): 53025–53042.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4): 541–551.

Li, H.; Li, S.; Dai, W.; Cao, M.; Kan, N.; Li, C.; Zou, J.; and Xiong, H. 2025. On disentangled training for nonlinear transform in learned image compression. *arXiv preprint arXiv:2501.13751*.

Li, H.; Li, S.; Dai, W.; Li, C.; Zou, J.; and Xiong, H. 2023. Frequency-aware transformer for learned image compression. *arXiv preprint arXiv:2310.16387*.

Lin, J.; Wang, K.; Wang, S.; Fan, S.; Li, G.; and Gao, W. 2025. VGD: Visual Geometry Gaussian Splatting for Feed-Forward Surround-view Driving Reconstruction. arXiv:2510.19578.

Liu, J.; Sun, H.; and Katto, J. 2023. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14388–14397.

Liu, Z.; Cheng, K.-T.; Huang, D.; Xing, E. P.; and Shen, Z. 2022. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation.

In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4942–4952.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Lu, H.; Liu, Q.; Liu, X.; and Zhang, Y. 2021a. A survey of semantic construction and application of satellite remote sensing images and data. *Journal of Organizational and End User Computing (JOEUC)*, 33(6): 1–20.

Lu, J.; Zhang, L.; Zhou, X.; Li, M.; Li, W.; and Gu, S. 2025. Learned Image Compression with Dictionary-based Entropy Model. *arXiv preprint arXiv:2504.00496*.

Lu, M.; Guo, P.; Shi, H.; Cao, C.; and Ma, Z. 2021b. Transformer-based image compression. *arXiv preprint arXiv:2111.06707*.

Ma, H.; Liu, D.; Yan, N.; Li, H.; and Wu, F. 2020. End-to-end optimized versatile image compression with wavelet-like transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3): 1247–1263.

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

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

Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Biderman, S.; Cao, H.; Cheng, X.; Chung, M.; Grella, M.; et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Roberts, A. W.; and Varberg, D. E. 1974. *Convex Functions: Convex Functions*, volume 57. Academic Press.

Sarvazyan, A. 1998. Mechanical imaging:: A new technology for medical diagnostics. *International journal of medical informatics*, 49(2): 195–216.

Strang, G. 1999. The discrete cosine transform. *SIAM review*, 41(1): 135–147.

Wallace, G. K. 1991. The JPEG still picture compression standard. *Communications of the ACM*, 34(4): 30–44.

Wang, K.; Yi, Q.; Ye, Y.; Li, S.; and Gao, W. 2025. AnyPcc: Compressing Any Point Cloud with a Single Universal Model. arXiv:2510.20331.

Witten, I. H.; Neal, R. M.; and Cleary, J. G. 1987. Arithmetic coding for data compression. *Communications of the ACM*, 30(6): 520–540.

Xie, L.; Gao, W.; and Zheng, H. 2022. End-to-End Point Cloud Geometry Compression and Analysis with Sparse Tensor. In *International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, 27–32.

Xie, Y.; Cheng, K. L.; and Chen, Q. 2021. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM international conference on multimedia*, 162–170.

Yang, F.; Herranz, L.; Cheng, Y.; and Mozerov, M. G. 2021. Slimmable compressive autoencoders for practical neural image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4998–5007.

Zeng, Y.; Guo, Y.; and Li, J. 2022. Recognition and extraction of high-resolution satellite remote sensing image buildings based on deep learning. *Neural Computing and Applications*, 34(4): 2691–2706.

Zhang, T.; Zhang, H.; Li, Y.; Li, L.; and Liu, D. 2025. Few-shot domain adaptation for learned image compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 10139–10147.

Zhu, Y.; Yang, Y.; and Cohen, T. 2022. Transformer-based transform coding. In *International conference on learning representations*.