JUSTC

Article

http://justc.ustc.edu.cn

Received: November 01, 2024; Accepted: May 20, 2025

The gap between principle and practice of lossy image coding

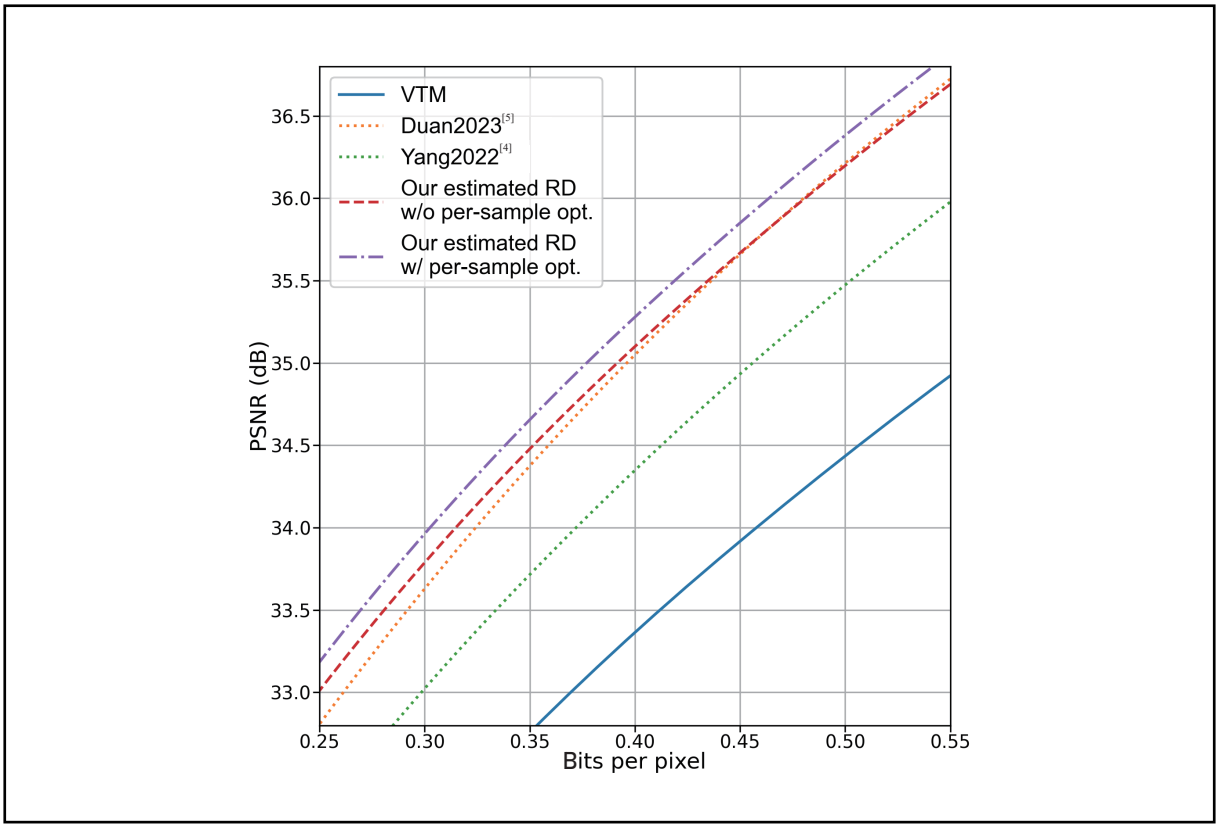
Haotian Zhang, and Dong Liu ✉

MOE Key Laboratory of Brain-Inspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei 230093, China

✉Correspondence: Dong Liu, E-mail: dongeliu@ustc.edu.cn

© 2026 The Author(s). This is an open access article under the CC BY-NC-ND 4.0 license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Graphical abstract



The estimated rate-distortion bounds of our method on the Kodak dataset.

Public summary

- This paper identifies the gap between the ideal rate-distortion function and the empirical rate-distortion function achieved by the state-of-the-art learned image coding schemes, revealing that the gap is incurred by five different effects: modeling effect, approximation effect, amortization effect, digitization effect, and asymptotic effect.
- This paper performs simulations and experiments to quantitatively evaluate the amortization effect, digitization effect, and asymptotic effect, through per-sample optimization techniques and comparative studies between analog and digital communication systems, respectively.
- This paper establishes a new rate-distortion bound of lossy image coding, achieving up to 35% rate savings over the H.266/VVC reference software on the Kodak dataset. This highlights the high potential of future lossy image coding technologies.

Citation: Zhang H T, Liu D. The gap between principle and practice of lossy image coding. *JUSTC*, **2026**, 56(1): 0101. CSTR: 32290.14.JUSTC-2024-0138. DOI: 10.52396/JUSTC-2024-0138

JUSTC

Article

http://justc.ustc.edu.cn

Received: November 01, 2024; Accepted: May 20, 2025

The gap between principle and practice of lossy image coding

Haotian Zhang, and Dong Liu ✉

MOE Key Laboratory of Brain-Inspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei 230093, China

✉Correspondence: Dong Liu, E-mail: dongeliu@ustc.edu.cn

© 2026 The Author(s). This is an open access article under the CC BY-NC-ND 4.0 license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Cite This: *JUSTC*, **2026**, 56(1): 0101 (11pp)

Read Online

Abstract: Lossy image coding is the art of computing that is principally bounded by the image's rate-distortion function. This bound, though never accurately characterized, has been approached practically via deep learning technologies in recent years. Indeed, learned image coding schemes allow direct optimization of the joint rate-distortion cost, thereby outperforming the handcrafted image coding schemes by a large margin. Still, it is observed that there is room for further improvement in the rate-distortion performance of learned image coding. In this article, we identify the gap between the ideal rate-distortion function forecasted by Shannon's information theory and the empirical rate-distortion function achieved by the state-of-the-art learned image coding schemes, revealing that the gap is incurred by five different effects: modeling effect, approximation effect, amortization effect, digitization effect, and asymptotic effect. We design simulations and experiments to quantitatively evaluate the last three effects, which demonstrates the high potential of future lossy image coding technologies.

Keywords: information theory; learned image coding; lossy image coding; rate-distortion function

CLC number: TN919.81 **Document code:** A

1 Introduction

Data compression, known as source coding in Shannon's information theory, aims to represent data into fewer symbols. Lossless source coding compresses discrete data without any loss of information. In contrast, lossy source coding achieves higher compression ratio by allowing for some loss of information, focusing on removing less important information. Due to limitations in storage and transmission capacity, lossy source coding is much more widely adopted in multimedia applications, such as audio, image, and video compression. In recent years, there has been rapid development of lossy source coding, primarily owing to deep learning technologies. However, the extent to how far current schemes are from the limit of compression performance remains insufficiently explored.

The rate-distortion function[1], defined as the minimum rate required to compress a data source given an expected distortion, establishes the fundamental limit on the compression efficiency of lossy source coding. Theoretically, any rate-distortion pairs above the rate-distortion function are achievable. However, calculating the rate-distortion function for real-world data sources with non-analytical distributions, such as natural images, poses significant challenges. Recent studies[2,3] have linked the Lagrangian-relaxed rate-distortion function to the optimization objectives of variational auto-encoders. Subsequent studies[4,5] have introduced upper-bound algorithms for estimating the rate-distortion function of generic sources, implying substantial potential for enhancing the performance of practical lossy source coding.

Moreover, the rate-distortion theory does not provide a direct method for designing lossy source coding schemes. Such design remains an art. It is true that the rate-distortion theory provides the guidance to design and optimize lossy source coding schemes, which aim to approach the rate-distortion function even though its exact form is still unknown. It is equally true that certain relaxations and assumptions are introduced during the design and optimization processes, incurring a gap between the principle and the practice of lossy source coding.

This paper aims to investigate the gap between the ideal performance forecasted by Shannon's information theory and the empirical performance achieved by lossy source coding schemes, with a focus on lossy image coding. The state-of-the-art lossy image coding schemes usually adopt a latent-based computing model, i.e., an image is converted to a set of features (aka latents) via the analysis process, and then the latents are entropy encoded into symbols; the symbols are entropy decoded to the latents, and then the latents are converted back to a reconstructed image via the synthesis process. For this paradigm, we identify that the principle-practice gap is incurred by five different effects:

- **Modeling effect.** Modeling effect comes from the model itself, specifically the entropy model used for the latents as well as the synthesis process.
- **Approximation effect.** Approximation effect comes from the restriction of the adopted computing architecture, i.e., the theoretically optimal computing model may not be found.
- **Amortization effect.** Amortization effect refers to the difference between optimizing the model parameters over the

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

entire training set and optimizing the model parameters for each sample individually.

- **Digitization effect.** Digitization effect refers to the difference between analog communications and digital communications, where the latter is assumed by the state-of-the-art lossy image coding. Digitization causes difficulty in obtaining the optimal model parameters.
- **Asymptotic effect.** Asymptotic effect reflects the fact that the theoretical rate is achievable in the asymptotic setting but practical schemes deal with finite-length images.

Following the analyses, we design simulations and experiments to quantitatively evaluate the last three effects. First, on top of a model trained by a large image set, we turn on per-sample optimization to further enhance the compression efficiency, which quantifies (at least) how large the gap is due to the amortization effect. Second, we estimate the theoretical achievable rate (as the mutual information) of an analog communication system and compare with the actual coding rate of the corresponding digital communication system, which quantifies how large the gap is due to the digitization and asymptotic effects. Our results demonstrate that the estimated rate-distortion function achieves 35% rate savings than the H.266/VVC reference software VTM on the Kodak image set, which sets up a new benchmark for lossy image coding. Accordingly, it reveals the high potential of future lossy image coding technologies.

2 Related work

2.1 Practices of lossy image coding

Lossy image coding is a fundamental problem in information theory. Over the past several decades, extensive research has focused on designing and optimizing practical lossy image coding schemes. In the following, we introduce the deterministic and stochastic practices separately.

2.1.1 Deterministic schemes

Most approaches rely on constructing deterministic systems that heavily rely on quantization to discard information from the source image data. A common framework followed by many methods is the transform coding scheme[6], where images are first transformed into a latent space to achieve decorrelation, followed by quantization and entropy coding.

In traditional lossy image codecs such as JPEG[7] and BPG [①], the different modules—transform, quantization, and entropy coding —are manually designed and optimized separately, rather than in an end-to-end manner.

In recent years, learned lossy image coding methods have achieved superior performance by leveraging the capabilities of neural networks. Unlike traditional approaches, learned lossy image coding[8] optimizes the various components in an end-to-end manner. While some methods[9,10] have explored vector quantization for end-to-end optimization, most studies adopted uniform scalar quantization, which currently represents the state-of-the-art approach in learned lossy image coding. Therefore, in the following, we focus specifically on methods based on uniform scalar quantization.

In the typical framework of learned image coding[8], the encoder applies an analysis transform to an input image x, producing a latent representation $g_a(x|\phi)$. The latent representation is then quantized to obtain the discrete latent $y = \lfloor g_a(x|\phi) \rceil$. This discrete latent is encoded losslessly using an entropy model $Q_Y(y|\psi)$. Finally, the decoder recovers y and reconstructs the image as $\hat{x} = g_s(y|\theta)$ using the synthesis transform. The parameters ϕ, θ, and ψ represent the trainable variables for the analysis transform, synthesis transform, and entropy model, respectively.

Lots of researchers have contributed to the design and optimization of learned lossy image coding methods, which can be primarily categorized into three types: the design of neural network-based transforms, the development of entropy models, and end-to-end optimization techniques.

The effectiveness of transforms plays a crucial role in approximating the optimal mapping from the source space to the latent space. Ballé et al.[8] introduced a convolutional neural network-based model for image compression. Additionally, more powerful convolutional neural networks, such as the residual connection and the attention mechanism, are adopted in Refs. [11, 12]. Some researchers[12–14] constructed transformer-based transforms. Additionally, instead of using non-invertible transforms, some studies utilized invertible transforms[15,16].

Entropy models are essential for estimating the distribution of latent variables. A factorized prior was employed by Ref. [8]. Ballé et al.[2] then proposed a hyperprior model that parameterizes the distribution as a Gaussian model conditioned on side information. Minnen et al.[17] introduced a more accurate entropy model that jointly utilizes a context model and hyperprior. Numerous studies[11,18–20] have contributed to enhancing entropy models for improved performance.

Since the gradient of the quantization operator is zero almost everywhere, standard back-propagation is inapplicable during training. Various quantization surrogates have been proposed to enable end-to-end optimization. Training with additive uniform noise[8] is a widely used method for approximating rounding. In Ref. [21], the straight-through estimator[22] was adopted for training, applying stochastic rounding in the forward pass while utilizing a modified gradient in the backward pass. The study of Ref. [18] empirically introduced a mixed quantization surrogate, which employs noisy latent for rate estimation but utilizes rounded latent and the straight-through estimator when passing through the synthesis transform. Additionally, the discrepancy in optimization objectives between training and testing, known as the train-test mismatch, has been investigated in studies such as Refs. [23–26].

2.1.2 Stochastic schemes

Some recent studies have attempted to establish stochastic lossy image coding systems to reduce train-test mismatches caused by quantization surrogates. In these systems, the transmitted latent is represented as $y = g_a(x|\phi) + \epsilon$, where ϵ is random noise. For instance, using uniform noise instead of rounding during testing is tried in Ref. [23], but its performance was found to be worse than using rounding[27,28]. Another

[①] http://bellard.org/bpg/

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

study[29] have experimented with Gaussian noise. Although these approaches based on Gaussian noise demonstrated promising theoretical ideal rate-distortion performance, the actual rate implemented may exhibit significant overhead[30], leading to suboptimal results.

2.2 Estimating rate-distortion function

Rate-distortion theory defines the rate-distortion function as the theoretical limit of lossy image coding. However, the calculation of this function for a general source is not specified in rate-distortion theory. Estimating the rate-distortion function for real-world sources is crucial for evaluating the potential improvements in practical lossy image codecs.

The classic Blahut-Arimoto algorithm[31,32] offers an optimization method for calculating the rate-distortion function of a discrete source with a known distribution. The study of Ref. [33] theoretically proposed a method for estimating the rate-distortion function based on empirically observed data samples. Recent advancements have incorporated deep learning techniques to estimate the rate-distortion function[4,5,34,35]. Notably, Ballé et al.[2] demonstrated a tight connection between the optimization of β-variational auto-encoder (β-VAE)[36] and the Lagrange-relaxed rate-distortion function. Furthermore, several studies[4,5] have established upper bounds for the rate-distortion function for natural images through training a β-VAE. Their findings indicate significant potential for enhancing existing lossy image codecs.

3 Theoretical analyses

The aforementioned image coding technologies all belong to the practice of lossy source coding. In Shannon's seminal work[1], lossy source coding was referred to as source coding with a fidelity criterion". For lossy source coding, Shannon defined the rate-distortion function as the theoretical limit of coding efficiency. However, rate-distortion theory does not specify how to design a lossy source coding system that achieves this function for a generic source, nor does it provide a method for calculating the rate-distortion function itself.

In practice, we can adhere to the principles of rate-distortion theory to design and optimize lossy source coding systems, with the hope that these systems will approach the rate-distortion function, even though its exact form remains unknown. With this in mind, we begin by reviewing learned image compression technologies, seeking to understand the gap between theoretical principles and practical implementation.

3.1 Formulation of optimization problem

In this paper, we use uppercase letters, such as Y, to denote random variables, while lowercase letters, such as y, represent specific samples of these random variables, without distinguishing between continuous and discrete variables. We represent the source by a random variable X that follows a distribution P_X, with the source alphabet denoted as $\mathcal{X}$. Similarly, we define $\hat{X}$ as the reconstruction, characterized by its distribution $P_{\hat{X}}$ and alphabet $\hat{\mathcal{X}}$. Furthermore, we assume that a distortion measure $\Delta: \mathcal{X}\times\hat{\mathcal{X}} \to \mathcal{R}_{\geqslant 0}$ has been established, such as the sum-of-squared-error defined by $\Delta(X,\hat{X}) = \|X-\hat{X}\|^2$. The notation $P_{\hat{X}|X}$ denotes a conditional distribution, which can also be interpreted as a virtual channel between X and $\hat{X}$.

The rate-distortion function $R(D)$ for the given source X and distortion measure $\Delta(\cdot,\cdot)$ is defined as

$$\begin{aligned} &R(D) = \inf_{P_{\hat{X}|X}} I(X;\hat{X}), \\ &\text{subject to } \mathbb{E}_{P_X P_{\hat{X}|X}}[\Delta(X,\hat{X})] \leqslant D, \end{aligned} \tag{1}$$

where $I(X;\hat{X})$ is the mutual information between X and $\hat{X}$, and $\mathbb{E}[\cdot]$ represents the mathematical expectation. The minimization of the mutual information is performed over all feasible $P_{\hat{X}|X}$ under the distortion constraint. The rate-distortion function is determined by the source distribution P_X and the pre-defined distortion measure $\Delta(\cdot,\cdot)$. This function describes the minimum achievable rate under a given distortion threshold, indicating that for any practical compression system, the rate cannot be less than $R(D)$ if the expected distortion is not larger than D.

Rate-distortion theory cannot be directly applied to evaluate the performance limits of lossy image coding or to design a practical codec. Due to various constraints like computational cost, some relaxations and assumptions are often introduced to solve Eq. (1) and develop lossy image codecs. These limitations contribute to a performance gap between the theoretical foundation and practical implementations of lossy image coding. By reviewing techniques in learned lossy image coding, we attribute this gap to five effects: modeling effect, approximation effect, amortization effect, digitization effect, and asymptotic effect, each of which will be introduced and discussed in the following.

3.1.1 Lagrangian relaxation

Since the rate-distortion function is convex, it can be reformulated as an unconstrained optimization problem to obtain the optimal solution. For any $\lambda > 0$, the unconstrained optimization objective is defined as

$$L(P_{\hat{X}|X}) = I(X;\hat{X}) + \lambda \mathbb{E}_{P_X P_{\hat{X}|X}}[\Delta(X,\hat{X})]. \tag{2}$$

For a given λ, a Pareto front of rate-distortion pairs can be achieved by minimizing Eq. (2). Each point on rate-distortion function curves can be reached by optimizing Eq. (2) through varying λ. However, if we consider optimizing another operational rate-distortion function problem, where constraints exist on the feasible region of $P_{\hat{X}|X}$, the operational rate-distortion function is not necessarily convex. In this scenario, Lagrangian methods can not reach some points on rate-distortion curves[37]. However, we can minimize Eq. (2) to find the convex hull of the operational rate-distortion function.

3.1.2 Modeling effect

The rate-distortion function can be estimated using the Monte Carlo method, which involves sampling from the data source,

$$\begin{aligned} I(X;\hat{X}) &= \mathbb{E}_{x\sim P_X}[KL(P_{\hat{X}|X=x}\|P_{\hat{X}})], \\ \mathbb{E}_{P_X P_{\hat{X}|X}}[\Delta(X,\hat{X})] &= \mathbb{E}_{x\sim P_X}\mathbb{E}_{P_{\hat{X}|X=x}}[\Delta(x,\hat{X})], \end{aligned} \tag{3}$$

where $KL(P_{\hat{X}|X=x}\|P_{\hat{X}})$ is the Kullback-Leibler divergence (relative entropy) between $P_{\hat{X}|X=x}$ and $P_{\hat{X}}$. Since $P_{\hat{X}}$ requires

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

knowledge of the source distribution, which in practice is usually unavailable. We need another probability measure $Q_{\hat{X}}$ to model the distribution of the reconstruction. It can be costly when dealing with high-dimensional sources with memory like natural images. Instead of directly modeling the distribution in the source or reconstruction space, practical lossy image coding methods usually adopt a latent variable model as shown in Fig. 1b for simplicity. In this framework, we need to map the source space into the latent space through $P_{Y|X}$, model the distribution of latent variable Y by Q_Y, and use another conditional distribution $P_{\hat{X}|Y}$ to synthesize reconstruction. In this framework, we have the objective as

$$\begin{aligned} L(P_{Y|X}, Q_Y, P_{\hat{X}|Y}) =& I(X;Y) + KL(P_Y \| Q_Y) + \\ & \lambda \mathbb{E}_{P_X P_{Y|X} P_{\hat{X}|Y}}[\Delta(X,\hat{X})], \\ I(X;Y) + KL(P_Y \| Q_Y) =& \mathbb{E}_{x\sim P_X}[KL(P_{Y|X=x} \| Q_Y)]. \end{aligned} \tag{4}$$

Eq. (4) is minimized it over all feasible $P_{Y|X}$, Q_Y, and $P_{\hat{X}|Y}$.

Some studies[38,39] introduced stochastic elements when synthesizing reconstruction for perception optimization. We consider the common scenario in learned image compression for distortion optimization, where $P_{\hat{X}|Y}$ is a deterministic transform, denoted as F_D. Then the objective becomes

$$\begin{aligned} L(P_{Y|X}, Q_Y, F_D) =& I(X;Y) + KL(P_Y \| Q_Y) + \\ & \lambda \mathbb{E}_{P_X P_{Y|X}}[\Delta(X, F_D(Y))]. \end{aligned} \tag{5}$$

Assuming we can obtain the optimal $P_{Y|X}$, we define the gap caused by the suboptimality of Q_Y, F_D as the modeling effect. The modeling effect consists of two components: the entropy modeling effect and the information loss effect:

- **Entropy modeling effect.** Entropy modeling effect comes from the inaccuracies in the chosen Q_Y when approximating P_Y, quantified as $KL(P_Y \| Q_Y)$. The gap is closed if $P_Y \equiv Q_Y$ such that $KL(P_Y \| Q_Y) = 0$.
- **Information loss effect.** Information loss effect results from the information loss during the synthesis process of reconstruction. Since the coding process is a Markov chain, $X \to Y \to \hat{X}$, we have $I(X;Y) \geqslant I(X;\hat{X})$, making the gap equal to $I(X;Y) - I(X;\hat{X})$. The gap is closed if F_D is bijective such that $I(X;Y) = I(X;\hat{X})$.

Many studies[2,11,14,17–20] focused on designing more effective entropy models to reduce the entropy modeling effect. Some studies[15,16] explored invertible neural networks to eliminate the information loss effect.

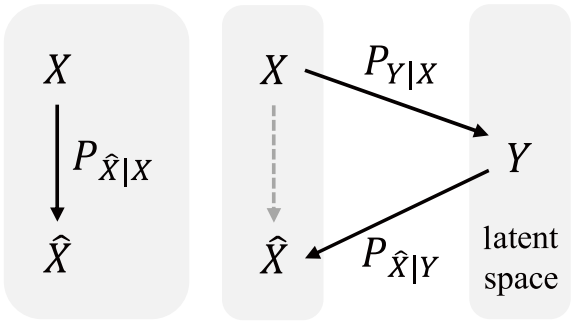


Fig. 1. (a) Illustration of a virtual channel $P_{\hat{X}|X}$ for lossy source coding; X is the source and $\hat{X}$ is the reconstruction. (b) Illustration of instantiating $P_{\hat{X}|X}$ by a latent-based model; Y refers to the latents.

3.1.3 Approximation effect

In this section, we focus on the optimization of $P_{Y|X}$ within the system described by Eq. (5). We denote the optimal $P_{Y|X}$ as $P^*_{Y|X}$. In practice, a specific form of $P_{Y|X}$ is typically assumed when optimizing Eq. (5). Previous studies usually use a basic form to approximate $P^*_{Y|X}$, such as factorized Gaussian or scalar quantization, primarily due to considerations of computational cost and sampling efficiency. These assumptions restrict the feasible region of $P_{Y|X}$, potentially leading to a suboptimal solution. We denote the restricted feasible solution set as A, and the optimal $P_{Y|X}$ within this subset as $P^{(A)*}_{Y|X}$. The performance gap between $P^*_{Y|X}$ and $P^{(A)*}_{Y|X}$ is referred to as the approximation effect[40].

For instance, most studies in learned lossy image coding treated $P_{Y|X}$ as a deterministic mapping that transforms the source space into a discrete latent space. We denote this deterministic mapping as F_E. Notably, unlike F_D, which can be a bijective transform, F_E cannot be bijective because it is designed to induce information loss to achieve lossy coding. In such a deterministic system, the objective becomes

$$\begin{aligned} L(F_E, Q_Y, F_D) =& H(Y) + KL(P_Y \| Q_Y) + \\ & \lambda \mathbb{E}_{P_X}[\Delta(X, F_D(F_E(X)))], \\ H(Y) + KL(P_Y \| Q_Y) =& \mathbb{E}_{P_X}[-\log Q_Y(F_E(X))]. \end{aligned} \tag{6}$$

Some studies treated $P_{Y|X}$ as a conditional uniform distribution[23], but its performance was found to be worse than using rounding[27,28]. Some studies[4,5,29] assumed $P_{Y|X}$ as a conditional Gaussian distribution, which shows the advancement for optimizing Eq. (5) compared to deterministic systems.

As discussed in Section 3.1.1, constraints on the feasible region of $P_{Y|X}$ may make the original optimization problem non-convex, such that optimizing the Lagrangian relaxation can only reach a subset in rate-distortion curves. Previous study[37] presented some toy examples where the operational rate-distortion function is not convex when Y is restricted to be discrete. For these sources, deterministic mappings can only address certain target rates.

3.1.4 Amortization effect

In this section, we explore the implementation of $P_{Y|X}$. To optimize Eq. (5), $P_{Y|X}$ is typically parameterized using neural networks, which are shared across the entire dataset, rather than directly optimizing Y for each sample x, which can be costly. For example, many studies parameterized the deterministic mapping described by Eq. (6) with scalar quantization as $Y = F_E(X) = \lfloor g_a(X|\phi) \rceil$, where ϕ represents the parameters that can be optimized. Some other studies parameterized the stochastic $P_{Y|X}$ as $Y = g_a(X|\phi) + \epsilon$, with ϵ denoting random noise. Theoretically, we could use a sufficiently large number of parameters to fit $P^{(A)*}_{Y|X}$ within our restricted set A. However, in practice, due to the limitations of neural networks and computational costs, $P^{(A)*}_{Y|X}$ may not be perfectly implemented. We refer to the performance gap between the implemented $P^{(A)}_{Y|X}$ and $P^{(A)*}_{Y|X}$ as the amortization effect[40]. The

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

term "amortization" highlights the difference arising from amortizing the parameters ϕ across the entire set of source samples rather than optimizing for each sample individually.

Many studies[13,14,19] attempted to adopt more powerful networks to fit $P_{Y|X}^{(A)*}$ to reduce this effect. An ensemble way is employed in Ref. [41] to enhance the ability of networks. There is another way[25,42–44] which further conducts per-sample optimization based on a well-optimized $P_{Y|X}^{(A)}$ to reduce the amortization effect. For $x \sim P_X$, we can optimize Y through

$$Y = \arg\min_{Y} KL(P_{Y|X=x} \| Q_Y) + \lambda \mathbb{E}_{P_{Y|X=x}}[\Delta(x, F_D(Y))]. \tag{7}$$

For a deterministic system described by Eq. (6), the objective can be simplified as

$$y = \arg\min_{y} -\log Q_Y(y) + \lambda \Delta(x, F_D(y)). \tag{8}$$

3.1.5 Digitization effect

For the systems established in the previous section, a critical challenge lies in the optimization techniques. With advancements in deep learning, gradient descent has become a widely adopted method for optimizing parameters. For continuous variables Y, unbiased path derivative gradient estimation can be efficiently implemented using the reparameterization trick[3]. However, for discrete variables Y, standard gradient descent is not applicable, because the derivative of the quantization operator is zero almost everywhere. The score function-based gradient estimator, also known as REINFORCE[45], provides an unbiased way to estimate the gradient for discrete Y. Nevertheless, due to the high latent dimensionality of natural images, this approach suffers from excessive variance. As a result, current implementations often rely on path derivative estimators.

We consider the deterministic system described by Eq. (6). There are two primary approaches to facilitate optimization using gradient descent. The first approach involves using a differentiable surrogate to replace the quantization operator, allowing for the application of standard gradient descent. However, this leads to an objective mismatch between the training and testing phases. The second approach is to define a derivative for the quantization operator, such as the identity function[22], which introduces gradient estimation error. This bias in gradient estimation cannot guarantee convergence of the objective function. The train-test mismatch and gradient estimation error during training for deterministic systems cause the digitization effect.

Uniform scalar quantization, with rounding as its common case, is widely adopted in learned image compression. To enable end-to-end optimization, various quantization surrogates have been proposed to replace rounding. Training with additive uniform noise[8] is a popular approach for approximating rounding. This method introduces stochasticity during training, which leads to a discrepancy between the training and testing phases. In Ref. [21], the straight-through estimator[22] is used, where stochastic rounding is applied during the forward pass, and a modified gradient is used in the backward pass. Several studies[18,24] have also tried to employ rounding in the forward pass and the straight-through estimator in the backward pass, avoiding train-test mismatch but introducing significant gradient estimation errors, which results in poor performance. Minnen and Singh[18] empirically proposed a mixed quantization surrogate, which uses noisy latent for rate estimation but applies rounded latent and the straight-through estimator during reconstruction synthesis. This mixed surrogate has demonstrated superior performance in several learned image compression models and has been widely adopted in recent state-of-the-art methods[13,14,20].

Due to the train-test mismatch inherent in the mixed quantization surrogate, optimizing the surrogate objective, denoted as $\widetilde{L}$, does not necessarily guarantee the minimization of the true objective L. Additionally, gradient estimation errors can lead to suboptimal solutions for $\widetilde{L}$. As a result, the performance of the obtained solution (F_E, Q_Y, F_D) may be significantly worse than that of the optimal solution for L.

The digitization effect also arises in per-sample optimization described in Eq. (8). Ref. [25] suggested employing an annealing proxy during per-sample optimization to reduce the digitization effect. By gradually decreasing the temperature coefficient, the proxy objective $\widetilde{L}$ converges toward the true objective L.

3.1.6 Asymptotic effect

The achievability of the rate-distortion function is established within the asymptotic setting, while the one-shot setting is standard in practical usage, thereby creating an asymptotic effect. In the asymptotic scenario, the average rate can approach mutual information as the dimension of the expanded source tends to infinity. In contrast, in the one-shot setting—where each sample is compressed individually—the rate may not be able to reach this lower bound.

In the systems described in Eq. (5), without any assumptions regarding the distributions of X and Y, previous studies[30,46] have proved a bound on the achievable rate in the one-shot setting, which is given by

$$I \leqslant R \leqslant I + \log(I+1) + O(1), \tag{9}$$

where $I = I(X;Y) + KL(P_Y \| Q_Y)$, and $O(1)$ represents a constant. The overhead $\log(I+1) + O(1)$ can be relatively small if the mutual information is large. However, it may be computationally expensive to communicate more information at once in practice[23]. For some specific distributions, this bound can be significantly tightened. For instance, if $P_{Y|X}$ is a conditional factorized uniform distribution, the rate can be achieved through universal quantization[47], which is bounded by $I \leqslant R \leqslant I + O(1)$, where $I = I(X;Y) + KL(P_Y \| Q_Y)$.

For a deterministic system described by Eq. (6), we only need to losslessly encode the discrete variable Y. The achievable rate in the one-shot setting satisfies $H \leqslant R \leqslant H + O(1)$, where $H = H(Y) + KL(P_Y \| Q_Y)$.

Some studies[29,48] have attempted to establish universal algorithms for communicating samples. These methods have also been investigated within the context of learned image compression[29]. Modeling $P_{Y|X}$ as a conditional Gaussian yields impressive results in estimating rate-distortion functions; however, performance in the one-shot setting remains inferior to that of deterministic systems. The overhead of the

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

rate caused by the asymptotic effect is substantial for such a system.

3.2 Advanced learned image coding methods

Learned lossy image coding methods have demonstrated significant performance improvements compared to traditional lossy image codecs. Among these methods, deterministic systems described in Eq. (6), have achieved state-of-the-art compression performance[13,14,20]. Ref. [28] theoretically highlights the superiority of deterministic systems in the one-shot setting. In this section, we examine the gap between this scheme and the theoretical limit.

We begin by analyzing how each module in the scheme contributes to the gap, including the analysis transform g_a, the synthesis transform g_s, the entropy model Q_Y, the quantization surrogate used during training (e.g., additive uniform noise), and the entropy coding engine (e.g., arithmetic coding). In this scheme, we have $F_E(x) = \lfloor g_a(x) \rceil$ and $F_D(y) = g_s(y)$. The quantization surrogate is employed during training for end-to-end optimization. In the encoding process, the analysis transform is applied to a source image x, producing a latent representation $g_a(x)$. The latent representation is then quantized to obtain the discrete latent $y = \lfloor g_a(x) \rceil$. This discrete latent is losslessly encoded into bitstreams by the entropy coding engine, utilizing the entropy model $Q_Y(y)$. In the decoding process, y is first losslessly recovered and then used to generate the reconstruction as $\hat{x} = g_s(y)$ through the synthesis transform.

The analysis transform primarily contributes to the amortization effect, digitization effect, and modeling effect. For a given g_s and Q_Y, the amortization effect is determined by the analysis transform. Employing a more powerful neural network as the analysis transform can reduce this gap. Additionally, per-sample optimization techniques, such as those proposed in Ref. [25, 44], can enhance the generalized analysis transform to further mitigate the amortization effect. The decorrelation capability of the analysis transform influences the modeling effect, since the optimization of the entropy model relies on the analysis transform. A more powerful analysis transform can alleviate the burden on the entropy model, thereby reducing the entropy modeling effect. The digitization effect is inevitable in systems that rely on quantization, particularly during the optimization for the analysis transform. Due to the presence of train-test mismatch and gradient estimation error, a more powerful analysis transform may not consistently reduce the amortization effect. In fact, employing a stronger analysis transform could potentially increase the digitization effect. Ref. [26] has shown some cases where utilizing more powerful networks as the analysis transform leads to degraded compression performance with certain quantization surrogates.

The synthesis transform primarily contributes to the modeling effect, amortization effect, and digitization effect. The function of the synthesis transform is to generate reconstructions from the latent variables and the optimal synthesis process is determined by the analysis transform. As discussed in Section 3.1.2, the modeling effect introduced by the synthesis transform could be eliminated by using an invertible network. In practice, due to the limited capabilities of the analysis transform and entropy model, the optimal synthesis process may take on a complex form. Employing a more powerful network as the synthesis transform can alleviate the burden on the analysis transform and the entropy model, thereby impacting both the amortization effect and the entropy modeling effect. Empirical results[15] demonstrate that using current invertible neural networks as synthesis transforms leads to suboptimal performance in practice. Consequently, researchers have opted for more powerful non-invertible neural networks to enhance overall performance. Regarding the digitization effect, the optimization of the synthesis transform is affected by the train-test mismatch. The extent of the influence of train-test mismatch is closely related to the characteristics of the synthesis transform[26].

The entropy model influences the modeling effect, amortization effect, and digitization effect. Enhancements in entropy modeling can reduce the entropy modeling effect through advanced structures like autoregressive models[17], latent variable models[2], flexible probabilistic models[8,11], and more powerful neural networks. Since the analysis transform and entropy model are jointly optimized, the ability of the entropy model also influences the amortization effect. Considering the digitization effect, similar to the synthesis transform, the optimization of the entropy model is affected by the train-test mismatch. The extent of the influence is related to the characteristics of the entropy model[26].

The quantization surrogate is a significant contributor to the digitization effect. Zhang et al.[26] indicate that the optimal choice of quantization surrogate differs among learned lossy image coding methods. Specifically, they highlight two critical elements influencing the effectiveness of quantization surrogates: train-test mismatch and gradient estimation error. This study indicates a tradeoff exists between the train-test mismatch and the gradient estimation risk, and the optimal balance varies across different network structures. However, how much the digitization effect affects compression performance is still not clearly understood. In per-sample optimization, the digitization effect remains, but it is simplified because the entropy model and synthesis transform are fixed, leaving only the optimization for a single sample. Yang et al.[25] proposed an annealing-based method to reduce this gap.

The entropy coding engine contributes to the asymptotic effect. Recent advanced methods usually use arithmetic coders[49] or asymmetric numeral systems[50] to encode symbols into bitstreams based on the entropy model. Since a large number of symbols are required to represent a natural image, the overhead of coding costs compared to the ideal rate is negligible.

In summary, in the current advanced deterministic schemes, the asymptotic effect is negligible, while the modeling effect, amortization effect, and digitization effect play crucial roles. To address the modeling and amortization effects, we can leverage more advanced networks or structures, such as large-scale generative models. This approach has demonstrated significant performance improvements in learned lossless image coding[51], where the digitization effect is not a concern. However, in lossy image coding, the digitization effect introduces significant uncertainty, complicating performance optimization and potentially limiting the

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

achievable compression performance.

3.3 Estimating rate-distortion function

To estimate an upper bound of the rate-distortion function for natural images, we focus only on minimizing Eq. (5) without the need to establish a practical implementation, without regarding the asymptotic effect. Studies[4,29] demonstrated the superior performance of continuous stochastic systems in optimizing this objective, without regarding the digitization effect. Additionally, a recent study[5] indicates that the estimated upper bound of the rate-distortion function can achieve up to 30% rate savings compared to VVC on the Kodak dataset, underscoring the potential of current lossy image coding methods.

When estimating this upper bound through Eq. (5) using a continuous stochastic system[4], we need to address the modeling effect, approximation effect, and amortization effect. As with deterministic systems, the modeling effect and amortization effect can be reduced by utilizing more powerful networks, and per-sample optimization can further mitigate the amortization effect. To address the approximation effect, a more effective distribution can be selected to approximate $P^*_{Y|X}$. Notably, factorized Gaussian distributions have shown impressive results for estimating the upper bound of the rate-distortion function for natural images, while more flexible distributions may further reduce the approximation effect.

4 Experimental analyses

In addition to theoretical analyses, we conduct experimental investigations to gain insights into the magnitude of the performance gap between practice and principle. We employ the upper-bound algorithm in Ref. [4] to estimate the ideal performance predicted by rate-distortion theory. Various structures and complexities are utilized to evaluate both the modeling effect and the amortization effect. By comparing the performance of the upper-bound algorithm with empirical performance in the one-shot setting involving quantization, we aim to provide some insights into the digitization effect.

4.1 Experimental setting

4.1.1 Learned image coding models

We employed two structures for our experiments: the mean-scale hyperprior model[17] and a model that incorporates more advanced entropy models[52]. All transform modules are constructed using partial convolution residual blocks[53], while depth-wise convolution residual blocks[54] are utilized in context models. We refer to the mean-scale hyperprior model with modified transforms as Res-hyper, and the version that integrates the advanced context model from Ref. [52] as Res-context. We adjust the modeling effect and amortization effect by adjusting the complexity of the neural networks. Specifically, we trained 8 Res-hyper models with various numbers of parameters by gradually increasing the depth and width of residual blocks in transforms. We trained one Res-context model by additionally introducing the context model on the largest Res-hyper model.

In Table 1, we present the configurations of our Res-hyper models with varying numbers of parameters. For the Res-context model, we incorporate the context model proposed in Ref. [52] into the largest Res-hyper model, following the same configurations. The key difference is that we expand the number of channels for latents to 512.

4.1.2 Training

We utilized mean squared error (MSE) in the RGB color space as the distortion metric, optimizing all models for this criterion. Multiple models were trained with varying values of $\lambda \in \{0.0018, 0.0054, 0.0162, 0.0483\}$. The training dataset contains approximately 20000 images from the Flicker2W dataset[55]. During each training iteration, images were randomly cropped into 256×256 patches.

For training, we employed the Adam optimizer[56] for 500 epochs, with a batch size of 8 and an initial learning rate of 5×10^{-5}. After 400 epochs, the learning rate was reduced to 2×10^{-5}, and after an additional 50 epochs, it was further decreased to 5×10^{-6}. A consistent random seed was utilized across all experiments.

In training deterministic systems, we trained with mixed quantization surrogates, zero-center quantization[18], and lower bounds for Gaussian scale parameters[26] for all models.

To estimate the upper bound of the rate-distortion function, we employed the Gaussian conditional distribution[4]. The Gaussian scale parameters were exponentially activated to avoid approaching zero. We calculated the estimated rate as

$$\begin{aligned}&\text{For } x\sim P_X, P_{Y|X=x}\sim\mathcal{N}(\mu,\sigma), Q_Y\sim\mathcal{N}(\hat{\mu},\hat{\sigma}),\\&R=KL(P_{Y|X=x}\|Q_Y)=\\&\left(\log\frac{\hat{\sigma}}{\sigma}+\frac{\sigma^2+(\mu-\hat{\mu})^2}{2\hat{\sigma}^2}-\frac{1}{2}\right)/\log(2)\text{ bits}.\end{aligned}\tag{10}$$

4.1.3 Per-sample optimization

Per-sample optimization was applied to reduce the amortization effect by optimizing all latent variables with the Adam optimizer. There are two latents in our model, the main latent Y and the hyperlatent Z, which characterizes the distribution of Y. Both Y and Z are optimized together. The initial learning rate was set to 5×10^{-3}, and it was gradually annealed using a cosine schedule to 10^{-4} over 5000 iterations. For optimizing deterministic systems, stochastic Gumbel annealing (SGA)[25] was employed to replace the rounding operator.

4.1.4 Evaluation

We evaluate the performance using the Kodak dataset①, which consists of 24 images with resolutions of either 512×768 or 768×512 pixels. To quantify rate and distortion, we use bits-per-pixel (bpp) and peak signal-to-noise ratio (PSNR), where PSNR is measured in the RGB color space. Additionally, we employ the BD-rate metric[57] to compute rate savings. For the overall BD-rate across the dataset, we calculate the BD-Rate for each image and then average these values. We also test the performance of VTM-22.0②, using the YUV444 input format for comparison.

4.2 Analyses

As illustrated in Fig. 2a, the performance of the estimated

①http://r0k.us/graphics/kodak/

②https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

Table 1. Configurations in our Res-hyper models with varying numbers of parameters.

Module	Name	Output size	Layer specification	3.2M	6.6M	8.8M	13.7M	21.1M	41.6M	89.3M
Analysis transform	down-sample1	$\frac{h}{2}\times\frac{w}{2}$	Channels[1]	32	48	48	64	96	128	128
	stage1	$\frac{h}{2}\times\frac{w}{2}$	Blocks[1]	1	1	2	2	2	3	3
			MLP ratio[1]	2	2	2	2	2	2	4
	down-sample2	$\frac{h}{4}\times\frac{w}{4}$	Channels	64	96	96	128	192	256	256
	stage2	$\frac{h}{4}\times\frac{w}{4}$	Blocks	1	1	2	2	2	3	3
			MLP ratio	2	2	2	2	2	2	4
	Down-sample3	$\frac{h}{8}\times\frac{w}{8}$	Channels	128	192	192	256	384	512	512
	stage3	$\frac{h}{8}\times\frac{w}{8}$	Blocks	3	3	6	6	6	9	9
			MLP ratio	2	2	2	2	2	2	4
	down-sample4	$\frac{h}{16}\times\frac{w}{16}$	Channels	256	384	384	512	512	512	512
	stage4	$\frac{h}{16}\times\frac{w}{16}$	Blocks	1	1	2	2	2	3	3
			MLP ratio	2	2	2	2	2	2	4
Hyper analysis transform	down-sample5	$\frac{h}{32}\times\frac{w}{32}$	Channels	192	256	256	256	384	512	512
	stage5	$\frac{h}{32}\times\frac{w}{32}$	Blocks	2	2	2	2	2	3	4
			MLP ratio	2	2	2	2	2	2	4
	down-sample6	$\frac{h}{64}\times\frac{w}{64}$	Channels	192	256	256	256	256	256	256

"Channels" denotes the number of output channels. "Blocks" denotes the number of residual blocks. "MLP ratio" indicates the expansion factor of hidden units in the multilayer perceptron (MLP) layers.

The structure of the synthesis transform and the hyper synthesis transform is symmetric to that of the analysis transform and hyper analysis transform, respectively.

bounds by the Res-hyper model improves as the number of parameters in the transform modules increases, although the marginal gains gradually diminish. The Res-context model, which uses a more advanced entropy model, shows better overall performance and larger marginal gains as the number of parameters continues to increase. Per-sample optimization, which enhances the analysis transform, consistently improves performance, but the gains decrease as model complexity increases, as illustrated in Fig. 3. By using more powerful transforms and entropy models, we can tighten the estimated upper bound of the rate-distortion function by reducing both the amortization and entropy modeling effects.

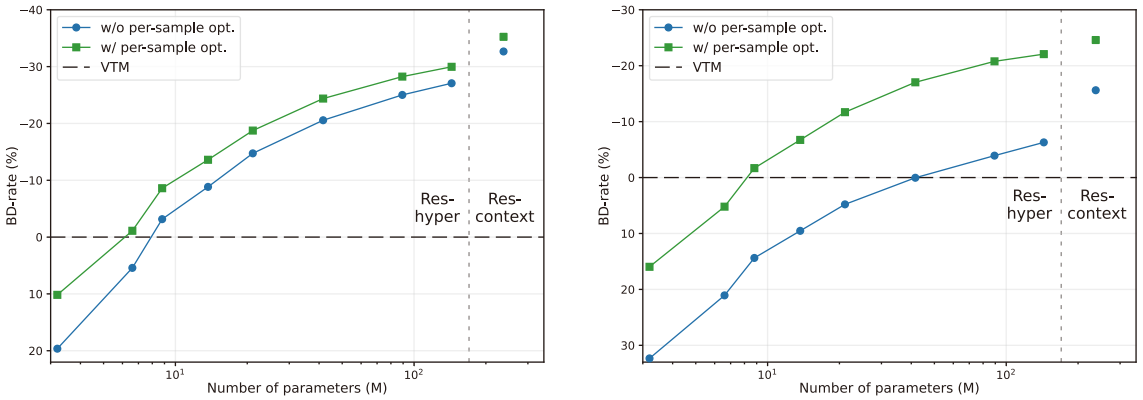
As shown in Fig. 2b, the empirical performance of the Res-hyper model in one-shot setting also improves as the number of parameters increases, but the speed of improvement is slower than that of the estimated bound, as depicted in Fig. 4. Furthermore, per-sample optimization proves to be more effective for enhancing empirical performance compared to the estimated bound, as illustrated in Fig. 3. Although per-sample optimization with the SGA surrogate is also affected by the digitization effect, it can reduce the gap introduced during training to some extent. These results provide insights into the magnitude of the digitization effect, which can be significant. The Res-context model shows better practical performance and larger marginal gains as the number of parameters continues to increase. An interesting phenomenon in Fig. 4 reveals that as the number of parameters increases, the gap between the estimated bound and the empirical performance for the Res-hyper model widens, while using the Res-context model results in a smaller gap with an increased number of parameters compared to the Res-hyper model. This suggests that the digitization effect may vary across different models.

Our experimental results also show that current practical learned image coding schemes can achieve significant performance improvements by employing more advanced transform modules, entropy models, and per-sample optimization. Specifically, these enhancements result in up to 24% rate savings compared to VTM on the Kodak dataset. As shown in Fig. 5, our results exhibit greatly enhanced performance, significantly outperforming current advanced models such as those reported in Refs. [13, 14].

Furthermore, we have tightened the estimated upper bound of the rate-distortion function, achieving up to 35% rate savings relative to VTM. Fig. 6 illustrates that our results present a tighter upper bound for the rate-distortion function of natural images compared to previous studies[4,5]. This highlights the high potential of future lossy image coding technologies.

When evaluating our model using MS-SSIM, the empirical rate-distortion function and the estimated rate-distortion function achieve rate savings of 22% and 36%, respectively, compared to VTM.

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101



(a) BD-rate of the estimated rate-distortion function over the performance of VTM

(b) BD-rate of the empirical rate-distortion function over the performance of VTM

Fig. 2. The compression efficiency of the estimated and empirical rate-distortion functions with different models and different numbers of parameters. A negative BD-rate indicates rate saving.

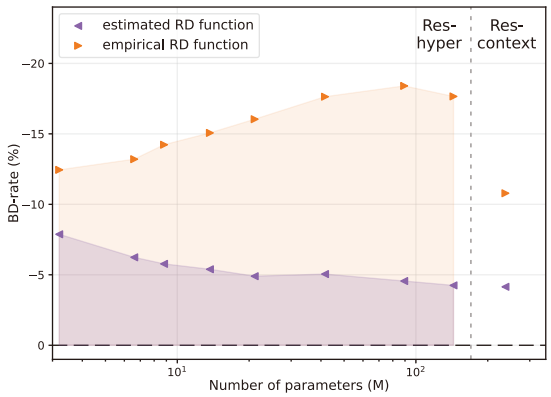


Fig. 3. The compression efficiency gain provided by the per-sample optimization on each model.

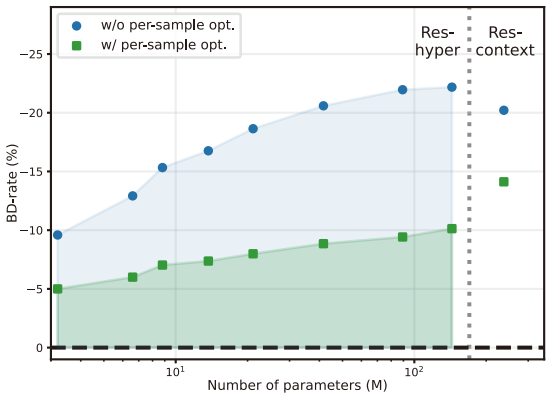


Fig. 4. The compression efficiency gap between the estimated and the empirical rate-distortion functions for different models and different numbers of parameters.

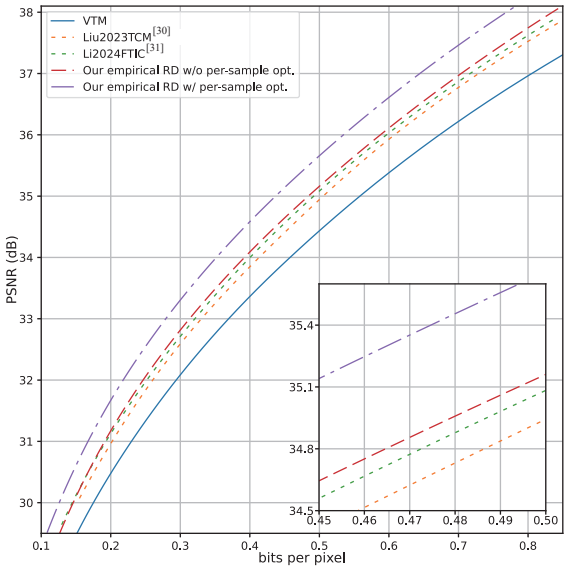


Fig. 5. Empirical rate-distortion performance of different methods. Our empirical RD shown in this figure is the performance based on the Res-context model.

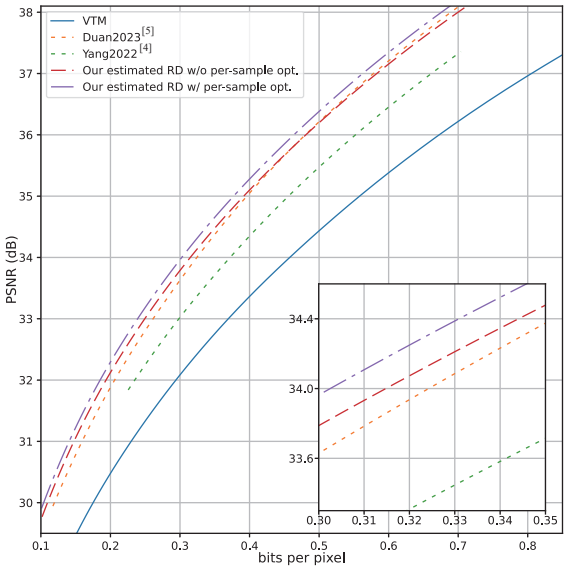


Fig. 6. Estimated upper bounds of rate-distortion functions. Our estimated RD shown in this figure is the performance based on the Res-context model.

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

5 Conclusions

In this paper, we analyze the gap between the ideal rate-distortion function and the empirical rate-distortion function for lossy image coding. We reveal that the gap is due to five different effects: modeling effect, approximation effect, amortization effect, digitization effect, and asymptotic effect. We analyze each effect by reviewing the technologies currently used in lossy image coding practices. Our simulation and experimental results illustrate how these effects impact compression performance. Additionally, we advance both the estimated and the empirical rate-distortion performance, demonstrating the high potential of future technologies.

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities (WK3490000006).

Conflict of interest

The authors declare that they have no conflict of interest.

Biographies

Haotian Zhang is currently pursuing the Ph.D. degree at the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC). He received the B.S. degree in Electronic Information Engineering from USTC in 2021. His research mainly focuses on image and video coding, machine learning, and computer vision.

Dong Liu is a Professor at the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC). He received the B.S. and Ph.D. degrees in Electrical Engineering from USTC in 2004 and 2009, respectively. His research mainly focuses on image and video processing, coding, analysis, and data mining.

References

[1] Shannon C E. Coding theorems for a discrete source with a fidelity criterion. In: Sloane N J A, Wyner A D, editors, Claude E. Shannon: Collected Papers, Wiley-IEEE Press, **1993**: 325–350.

[2] Ballé J, Minnen D, Singh S, et al. Variational image compression with a scale hyperprior. In: 6th International Conference on Learning Representations. Vancouver, BC, Canada: ICLR, **2018**.

[3] Kingma D P, Welling M. Auto-encoding variational Bayes. arXiv: 1312.6114, **2013**.

[4] Yang Y, Mandt S. Towards empirical sandwich bounds on the rate-distortion function. arXiv: 2111.12166, **2021.**

[5] Duan Z, Ma J, He J, et al. An improved upper bound on the rate-distortion function of images. In: 2023 IEEE International Conference on Image Processing (ICIP). Kuala Lumpur, Malaysia: IEEE, **2023**: 246–250.

[6] Goyal V K. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, **2001**, *18*: 9–21.

[7] Wallace G K. The JPEG still picture compression standard. *Communications of the ACM*, **1991**, *34*: 30–44.

[8] Ballé J, Laparra V, Simoncelli E P. End-to-end optimized image compression. In: International Conference on Learning Representations (ICLR), ICLR, **2017**.

[9] Feng R, Guo Z, Li W, et al. NVTC: Nonlinear vector transform coding. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, **2023**: 6101–6110.

[10] Zhang X, Wu X. LVQAC: Lattice vector quantization coupled with spatially adaptive companding for efficient learned image compression. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, **2023**: 10239–10248.

[11] Cheng Z, Sun H, Takeuchi M, et al. Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, **2020,** 7936–7945.

[12] Zhu Y, Yang Y, Cohen T. Transformer-based transform coding. In: International Conference on Learning Representations, **2022**.

[13] Liu J, Sun H, Katto J. Learned image compression with mixed transformer-CNN architectures. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, **2023**: 14388–14397.

[14] Li H, Li S, Dai W, et al. Frequency-aware transformer for learned image compression. In: International Conference on Learning Representation (ICLR), **2025**.

[15] Ma H, Liu D, Yan N, et al. End-to-end optimized versatile image compression with wavelet-like transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2022**, *44*: 1247–1263.

[16] Helminger L, Djelouah A, Gross M, et al. Lossy image compression with normalizing flows. arXiv: 2008.10486, **2020**.

[17] Minnen D, Minnen D, Ballé J, et al. Joint autoregressive and hierarchical priors for learned image compression. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. New York: ACM, **2018**: 10794–10803.

[18] Minnen D, Singh S. Channel-wise autoregressive entropy models for learned image compression. In: 2020 IEEE International Conference on Image Processing (ICIP). Abu Dhabi, United Arab Emirates: IEEE, **2020**: 3339–3343.

[19] He D, Yang Z, Peng W, et al. ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, **2022**: 5708–5717.

[20] Jiang W, Jiang W, Yang J, et al. MLIC: Multi-reference entropy model for learned image compression. In: Proceedings of the 31st ACM International Conference on Multimedia. New York: ACM, **2023**: 7618–7627.

[21] Theis L, Shi W, Cunningham A, et al. Lossy image compression with compressive autoencoders. arXiv: 1703.00395, **2017**.

[22] Bengio Y, Léonard N, Courville A. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv: 1308.3432, **2013**.

[23] Agustsson E, Theis L. Universally quantized neural compression. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: ACM, **2020**: 12367–12376.

[24] Guo Z, Zhang Z, Feng R, et al. Soft then hard: Rethinking the quantization in neural image compression. arXiv: 2104.05168, **2021**.

[25] Yang Y, Yang Y, Bamler R, et al. Improving inference for neural image compression. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: ACM, **2020**: 573–584.

[26] Zhang H, Li L, Liu D. On uniform scalar quantization for learned image compression. arXiv: 2309.17051, **2023**.

[27] Balle J, Chou P A, Minnen D, et al. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, **2021**, *15* (2): 339–353.

[28] Theis L, Agustsson E. On the advantages of stochastic encoders. arXiv: 2102.09270, **2021**.

[29] Flamich G, Havasi M, Hernández-Lobato J M. Compressing images by encoding their latent representations with relative entropy coding. In: Proceedings of the 34th International Conference on Neural

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101

Information Processing Systems, New York: ACM, **2020**: 16131–16141.
[30] Li C T, El Gamal A. Strong functional representation lemma and applications to coding theorems. *IEEE Transactions on Information Theory*, **2018**, *64*: 6967–6978.
[31] Blahut R. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, **1972**, *18*: 460–473.
[32] Arimoto S. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, **1972**, *18*: 14–20.
[33] Harrison M T, Kontoyiannis I. Estimation of the rate–distortion function. *IEEE Transactions on Information Theory*, **2008**, *54*: 3757–3762.
[34] Lei E, Hassani H, Saeedi Bidokhti S. Neural estimation of the rate-distortion function with applications to operational source coding. *IEEE Journal on Selected Areas in Information Theory*, **2022**, *3*: 674–686.
[35] Yang Y, Yang Y, Eckstein S, et al. Estimating the rate-distortion function by Wasserstein gradient descent. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New York: ACM, **2023**: 2768–2794.
[36] Higgins I, Matthey L, Pal A, et al. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations. ICLR, **2017**.
[37] Wagner A B, Ballé J. Neural networks optimally compress the sawbridge. In: 2021 Data Compression Conference (DCC). Snowbird, UT, USA: IEEE, **2021**: 143–152.
[38] Agustsson E, Tschannen M, Mentzer F, et al. Generative adversarial networks for extreme learned image compression. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea: IEEE, **2019**: 221–231.
[39] Yang R, Mandt S. Lossy image compression with conditional diffusion models. arXiv: 2209.06950, **2022**.
[40] Cremer C, Li X, Duvenaud D. Inference suboptimality in variational autoencoders. In: Proceedings of the 35th International Conference on Machine Learning. PMLR, **2018**, *80*: 1078–1086.
[41] Wang Y, Liu D, Ma S, et al. Ensemble learning-based rate-distortion optimization for end-to-end image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, **2021**, *31*: 1193–1207.
[42] Kim Y, Wiseman S, Miller A C, et al. Semi-amortized variational autoencoders. In: Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden: PMLR, **2018**, 80: 2678–2687.
[43] Marino J, Yue Y, Mandt S. Iterative amortized inference. In: Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden: PMLR, **2018**, *80*: 3403–3412.
[44] Campos J, Meierhans S, Djelouah A, et al. Content adaptive optimization for neural image compression. arXiv: 1906.01223, **2019**.
[45] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, **1992**, *8*: 229–256.
[46] Li C T, Anantharam V. A unified framework for one-shot achievability via the Poisson matching lemma. *IEEE Transactions on Information Theory*, **2021**, *67* (5): 2624–2651.
[47] Ziv J. On universal quantization. *IEEE Transactions on Information Theory*, **1985**, *31*: 344–347.
[48] Lucas Theis L, Ahmed N Y. Algorithms for the communication of samples. In: Proceedings of the 39th International Conference on Machine Learning, PMLR, **2022**, *162*: 21308–21328.
[49] Langdon G G. An introduction to arithmetic coding. *IBM Journal of Research and Development*, **1984**, *28*: 135–149.
[50] Duda J. Asymmetric numeral systems: Entropy coding combining speed of Huffman coding with compression rate of arithmetic coding. arXiv: 1311.2540, **2013**.
[51] Li Z, Huang C, Wang X, Hu H, et al. Understanding is compression. arXiv: 2407.07723, **2024**.
[52] Li J, Li B, Lu Y. Neural video compression with diverse contexts. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, **2023**: 22616–22626.
[53] Chen J, Kao S H, He H, et al. Run, don't walk: Chasing higher FLOPS for faster neural networks. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, **2023**: 12021–12031.
[54] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: IEEE, **2017**: 1800–1807.
[55] Liu J, Lu G, Hu Z, et al. A unified end-to-end framework for efficient deep image compression. arXiv: 2002.03370, **2020**.
[56] Kingma D P, Ba L J. Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR), Ithaca, NY, USA: ArXiv, **2015**.
[57] Bjontegaard G. Calculation of average PSNR differences between RD-curves. Technical report. Austin, TX, USA: Document VCEG-M33, **2001**.

CSTR: 32290.14.JUSTC-2024-0138
DOI: 10.52396/JUSTC-2024-0138
JUSTC, **2026**, 56(1): 0101