

# Conditional Latent Coding with Learnable Synthesized Reference for Deep Image Compression

Siqi Wu<sup>1\*</sup>, Yinda Chen<sup>2†</sup>, Dong Liu<sup>2</sup>, Zhihai He<sup>3‡</sup>

<sup>1</sup>University of Missouri, Columbia, MO, USA

<sup>2</sup>University of Science and Technology of China, Hefei, China

<sup>3</sup>Southern University of Science and Technology, Shenzhen, China

## Abstract

In this paper, we study how to synthesize a dynamic reference from an external dictionary to perform conditional coding of the input image in the latent domain and how to learn the conditional latent synthesis and coding modules in an end-to-end manner. Our approach begins by constructing a universal image feature dictionary using a multi-stage approach involving modified spatial pyramid pooling, dimension reduction, and multi-scale feature clustering. For each input image, we learn to synthesize a conditioning latent by selecting and synthesizing relevant features from the dictionary, which significantly enhances the model’s capability in capturing and exploring image source correlation. This conditional latent synthesis involves a correlation-based feature matching and alignment strategy, comprising a Conditional Latent Matching (CLM) module and a Conditional Latent Synthesis (CLS) module. The synthesized latent is then used to guide the encoding process, allowing for more efficient compression by exploiting the correlation between the input image and the reference dictionary. According to our theoretical analysis, the proposed conditional latent coding (CLC) method is robust to perturbations in the external dictionary samples and the selected conditioning latent, with an error bound that scales logarithmically with the dictionary size, ensuring stability even with large and diverse dictionaries. Experimental results on benchmark datasets show that our new method improves the coding performance by a large margin (up to 1.2 dB) with a very small overhead of approximately 0.5% bits per pixel.

**Code** — <https://github.com/ydchen0806/CLC>.

## 1 Introduction

With the rapid development of the Internet and mobile devices, billions of images are available in the world. For a given image, it is easy to find many correlated images on the Internet. It will be very interesting to explore how to utilize this vast amount of data to establish a highly efficient representation of the input image to improve the performance of deep image compression. Continuous efforts have been

\*This work was completed during Siqi Wu’s visiting research period at Southern University of Science and Technology.

†Co-first author.

‡Corresponding author. Email: hezh@sustech.edu.cn.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

made in the past two decades. The early attempt is to extract low-level feature patches from external images as a dictionary for image super-resolution (Sun et al. 2003) and quality enhancement (Xiong, Sun, and Wu 2010). Yue *et al.* (Yue et al. 2013) proposed a cloud-based image coding scheme that utilizes a large-scale image database for reconstruction, achieving high compression ratios while maintaining visual quality. As data compression has shifted to the deep image/video compression paradigm in recent years, we would like to explore how to utilize the external dictionary of images to generate a dynamic reference representation to perform conditional coding of the input image within the deep image compression framework.

Deep neural network-based image compression methods (Ballé, Laparra, and Simoncelli 2017; Toderici et al. 2017; Lee, Cho, and Beack 2019) have made significant progress in recent years, surpassing traditional transform coding methods like JPEG in compression efficiency. However, current deep learning compression still faces challenges in efficiently exploring the source correlation of the image and maintaining high reconstruction quality at low bit rates. To further improve compression efficiency, researchers have begun to explore the use of external images as side information in distributed deep compression. For example, Ayzik *et al.* (Ayzik and Avidan 2020) used auxiliary image information to perform block matching in the image domain, while Huang *et al.* (Huang et al. 2023) extended this concept by introducing a multi-scale patch matching approach. However, this approach relies on specific auxiliary images, limiting its applicability and improvement.

To overcome these limitations, we propose a novel framework called Conditional Latent Coding (CLC), which uses auxiliary information as a conditional probability at both the encoder and decoder. Our approach constructs a universal image feature dictionary using a multi-stage process involving modified spatial pyramid pooling (SPP), dimensionality reduction, and multi-scale feature clustering. For each input image, we generate a conditioning latent by adaptively selecting and learning to combine relevant features from the dictionary to generate a highly efficient reference representation, called *conditioning latent*, for the input image. We then apply an advanced feature matching and alignment strategy, comprising a Conditional Latent Matching (CLM) module and a Conditional Latent Synthesis (CLS) module.

This process leverages the conditioning latent to guide the encoding process, allowing for more efficient compression by exploiting similarities between the input image and the reference features. As demonstrated in our experimental results on benchmark datasets, our new method improves the coding performance by a large margin (up to 1.2 dB) at low bit-rates.

## 2 Related Work and Unique Contributions

Deep learning-based image compression has achieved remarked progress in recent years. Ballé *et al.* (Ballé, Laparra, and Simoncelli 2017) pioneered an end-to-end optimizable architecture, later enhancing it with a hyperprior model (Ballé *et al.* 2018) to improve entropy estimation. Transformer architectures have been proposed by Qian *et al.* (Qian *et al.* 2022) to improve probability distribution estimation. Similarly, Cheng *et al.* (Cheng *et al.* 2020) parameterizes the distributions of latent codes with discretized Gaussian Mixture models. Liu *et al.* (Liu, Sun, and Katto 2023) combined CNNs and Transformers in the TCM block to explore the local and non-local source correlation. Yang *et al.* (Yang 2023) proposed a Tree-structured Implicit Neural Compression (TINC) to maintain the continuity among regions and remove the local and non-local redundancy. To enhance the entropy coding performance, the conditional probability model and joint autoregressive and hierarchical priors model have been developed in (Mentzer *et al.* 2018; Minnen, Ballé, and Toderici 2018). Jia *et al.* (Jia *et al.* 2024) introduced a Generative Latent Coding (GLC) architecture to achieve high-realism and high-fidelity compression by transform coding in the latent space.

This work is related to reference-based deep image compression, where reference information is used to improve coding efficiency. For example, Li *et al.* (Li, Li, and Lu 2021) pioneered this approach in video compression, while Ayzik *et al.* (Ayzik and Avidan 2020) applied it at the decoder level. Sheng *et al.* (Sheng *et al.* 2022) proposed a temporal context mining module to propagate features and learn multi-scale temporal contexts. Huang *et al.* (Huang *et al.* 2023) extended the concept to multi-view image compression with advanced feature extraction and fusion. Li *et al.* (Li, Li, and Lu 2023) introduced the group-based offset diversity to explore the image context for better prediction. Zhao *et al.* (Zhao *et al.* 2021) optimized the reference information using a universal rate-distortion optimization framework. (Zhao *et al.* 2023) integrated side information optimization with latent optimization to further enhance the compression ratio. In (Li *et al.* 2023), within the context of underwater image compression, a multi-scale feature dictionary was manually created to provide a reference for deep image compression based on feature matching. A content-aware reference frame selection method was developed in (Wu *et al.* 2022) for deep video compression.

**Unique contributions.** In comparison to existing methods, our work has the following unique contributions. (1) We develop a new approach, called conditional latent coding (CLC), which learns to synthesize a dynamic reference for each input image to achieve highly efficient conditional

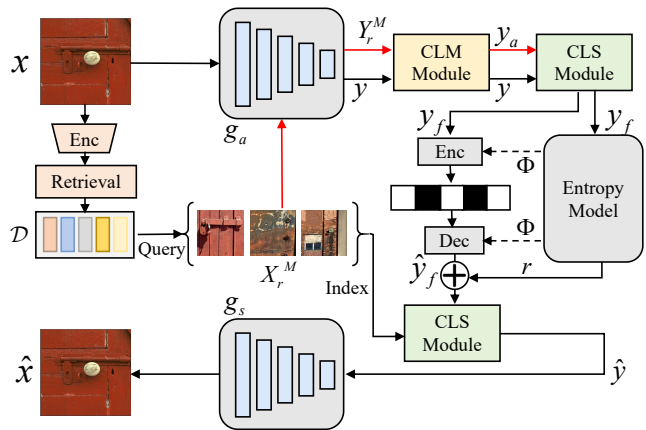


Figure 1: Overview of the proposed Conditional Latent Coding (CLC) framework.

coding in the latent domain. (2) We develop a fast and efficient feature matching scheme based on ball tree search and an effective feature alignment strategy that dynamically balances compression bit-rate and reconstruction quality. (3) We developed a theoretical analysis to show that the proposed CLC method is robust to perturbations in the external dictionary samples and the selected conditioning latent, with an error bound that scales logarithmically with the dictionary size, ensuring stability even with large and diverse dictionaries.

## 3 The Proposed CLC Method

### 3.1 Method Overview

The overall architecture of our proposed CLC framework is illustrated in Figure 1. Given an input image  $x$ , we first construct a pre-trained feature reference dictionary  $D$  from a large reference image dataset using a multi-stage approach involving feature extraction with modified spatial pyramid pooling (SPP), dimensionality reduction, and multi-scale feature clustering. Then, given an input image  $x$ , we extract its feature using an encoder  $F_\theta$  which is used to query the dictionary  $D$  and find the top  $M$  best-matching reference images  $X_r^M = \{x_r^1, x_r^2, \dots, x_r^M\}$ . In this work, the default value of  $M$  is 3. Both  $x$  and the queried reference  $X_r^M$  are passed through the encoder transform network  $g_a$  to obtain their latent representations  $y$  and  $Y_r^M$ , respectively. Using  $Y_r^M$  as reference, we obtain  $y_f$  through adaptive feature matching and multi-scale alignment and then learn a network to perform conditional latent coding of  $y_f$ . Simultaneously, a hyperprior network  $h_a$  estimates a hyperprior  $z$  from  $y_f$  to provide additional context for entropy estimation. A slice-based autoregressive context model is used for entropy coding, dividing  $y_f$  into slices and using both  $z$  and previously coded elements to estimate probabilities. During decoding, we first reconstruct  $z$  and  $y_f$  from the bitstream, then use the dictionary indices passed from the encoder to apply the same reference processing and alignment procedure to reconstruct  $y$  from  $y_f$ , and finally reconstruct the image  $\hat{x}$  using the synthesis transform  $g_s$ . In the following

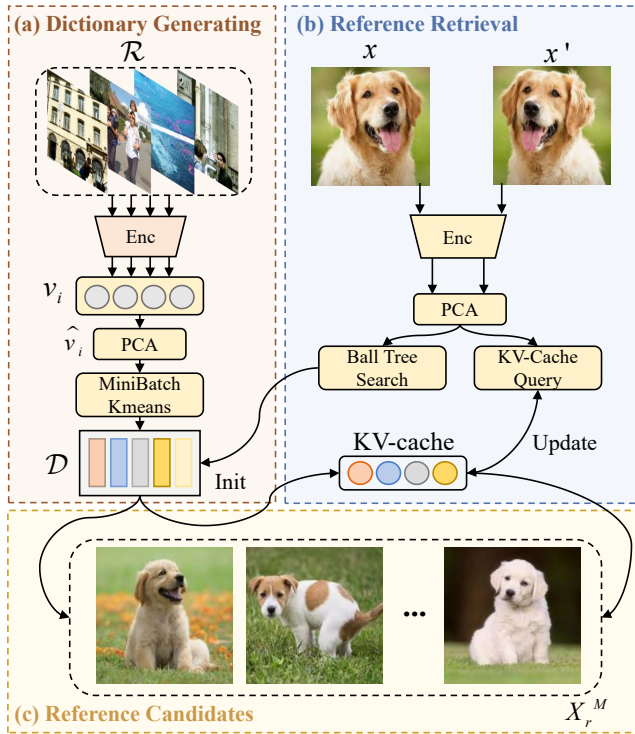


Figure 2: Universal Feature Dictionary Construction. (a) Dictionary Generation using diverse images  $\mathcal{R}$  to create initial  $\mathcal{D}$ . (b) Reference Retrieval for querying and updating dictionary with inputs  $x$  and  $x'$ . (c) Examples of reference candidates  $X_r^M$  retrieved from the dictionary.

section, we will explain the proposed CLC method in more detail.

### 3.2 Constructing the Support Dictionary

As stated in the above section, our main idea is to construct a universal feature dictionary from which a reference latent can be dynamically generated to perform conditional latent coding of each image. Here, a critical challenge is constructing a universal feature dictionary that effectively represents diverse image content and enables efficient feature utilization throughout the compression pipeline. We address this challenge using a multi-stage approach that combines advanced feature extraction, dimensionality reduction, feature clustering, and fast and efficient dictionary access by the deep image compression system, as illustrated in Figure 2.

**(1) Constructing the reference feature dictionary.** Our method begins with a large reference dataset  $\mathcal{R} = \{x_1, x_2, \dots, x_N\}$ . In this work, we randomly download 3000 images from the web. We use a modified pre-trained ResNet-50 model with Spatial Pyramid Pooling (SPP) as our feature extractor. For each image  $x_i$ , we extract its feature  $\mathbf{v}_i = \text{SPP}(f_\theta(x_i))$ , where  $f_\theta(\cdot)$  represents the ResNet-50 backbone, and SPP aggregates features at scales  $\{1 \times 1, 2 \times 2, 4 \times 4\}$ . This multi-scale approach captures both global and local image characteristics.

To manage the high dimensionality of these features, we apply Principal Component Analysis (PCA), reducing each vector to 256 dimensions  $\hat{\mathbf{v}}_i$ . The reduced feature set is then clustered using MiniBatch K-means, yielding  $K$  clusters:  $\{C_1, C_2, \dots, C_K\}$ . From each cluster  $C_j$ , we select the feature vector closest to the centroid as its representative:  $\mathbf{d}_j = \arg \min_{\hat{\mathbf{v}} \in C_j} \|\hat{\mathbf{v}} - \boldsymbol{\mu}_j\|_2$ , where  $\boldsymbol{\mu}_j$  is the centroid of  $C_j$ . These representatives form our feature dictionary  $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_K\}$ .

**(2) Fast and efficient dictionary matching.** Our proposed CLC method deep image compression needs to access this dictionary during training and inference. One central challenge here is the dictionary search and matching efficiency. For efficient feature dictionary management and access, we introduce a KV-cache mechanism that is employed in both the initial feature retrieval and the subsequent encoding-decoding process. Specifically, we define our KV-cache as a tuple  $(\mathbf{K}, \mathbf{V})$ , where  $\mathbf{K} \in \mathbb{R}^{N \times d_k}$  represents the keys and  $\mathbf{V} \in \mathbb{R}^{N \times d_v}$  represents the values. Here,  $N$  is the number of entries in the cache,  $d_k$  is the dimension of the keys, and  $d_v$  is the dimension of the values.

In the feature retrieval phase, we construct a ball tree over  $\mathcal{D}$  for the initial coarse search, while maintaining the KV-cache. During compression, given an input image  $x$ , we extract its feature  $f_\theta(x)$  and use it to query both the Ball Tree and the KV-cache. The retrieval process is formulated as a scaled dot-product attention mechanism:

$$A(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left( \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \right) \mathbf{V}, \quad (1)$$

where  $\mathbf{Q} = f_\theta(x)$ , and  $\mathbf{K}$  and  $\mathbf{V}$  are the keys and values in the KV-cache, respectively. To manage the size of the KV-cache and improve the matching efficiency, we implement a compression technique. Let  $C: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$  be our compression function, where  $d' < d$ . We apply this to both keys and values:

$$\mathbf{K}_c = C(\mathbf{K}), \quad \mathbf{V}_c = C(\mathbf{V}). \quad (2)$$

The compression function  $C$  is designed to preserve the most important information while reducing the dimensionality. In practice, we implement  $C$  as a learnable neural network layer, optimized jointly with the rest of the system. Furthermore, to enhance the efficiency of our KV-cache, we implement an eviction strategy  $E: \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{N' \times d}$ , where  $N' < N$ . This strategy removes less useful entries from the cache based on a relevance metric  $\rho: \mathbb{R}^d \rightarrow \mathbb{R}$ :

$$(\mathbf{K}_e, \mathbf{V}_e) = E(\mathbf{K}, \mathbf{V}) = \text{TopK}(\rho(\mathbf{K}_i), \mathbf{K}, \mathbf{V}), \quad (3)$$

where TopK selects the top  $K$  entries based on the relevance scores. To further enhance robustness, we implement a multi-query strategy. For an input image  $x$ , we generate an augmented version  $x'$  (e.g., by rotation) and perform separate queries for both. The final set of reference features is obtained by merging and de-duplicating the results.

### 3.3 Conditional Latent Synthesis and Coding

As the unique contribution of this work, instead of simply finding the best match in existing methods (Jia et al.

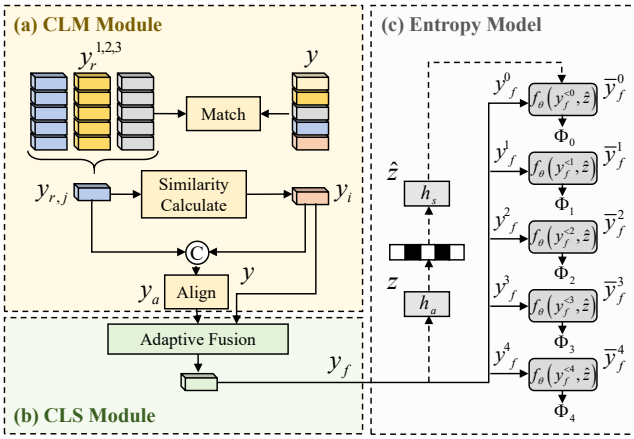


Figure 3: The detail of our proposed CLM and CLS module.

2024), the reference or side information for each image is dynamically generated in the latent domain by a learned network to best represent the input image. Our method is motivated by the following observation: the central challenge in reference-based image compression is the large deviation between the arbitrary input image and the fixed and limited set of reference images. Our method finds multiple closest reference images and dynamically fuses them to form a best approximation of the input image in the latent domain. Specifically, the proposed conditional latent synthesis and coding method has the following major components:

**(1) Feature Matching and Alignment.** We first propose an advanced feature matching and alignment scheme that aligns reference features from the dictionary with the input image. Our approach begins with a Conditional Latent Matching (CLM) module. Given an input image  $x \in \mathbb{R}^{H \times W \times 3}$  and a pre-built feature reference dictionary  $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_K\}$ , we first extract features from  $x$  to query  $\mathcal{D}$ , retrieving the top  $M$  feature and their corresponding reference images  $X_r^M = \{x_r^1, x_r^2, \dots, x_r^M\}$ . Both  $x$  and  $X_r^M$  are then processed through the same analysis transform network. In this work, we use the Transformer-CNN Mixture (TCM) block (Liu, Sun, and Katto 2023), which efficiently combines the strengths of CNNs for local feature extraction and transformers for capturing long-range dependencies. TCM blocks are used at both encoder  $g_a$ , decoder  $g_s$ , and hyperprior network  $h_a$ , enabling effective feature processing at various stages of the compression pipeline.

The analysis transform  $g_a$  converts  $x$  and  $x_r^M$  into latent representations  $y$  and  $Y_r^M$ , respectively. The CLM then establishes correspondences between  $Y_r^M$  and  $y$ , addressing the issues of spatial inconsistencies. It computes  $y_m = \mathcal{F}_m(y, Y_r^M; \theta_m)$ , where  $\mathcal{F}_m$  is a learnable function parameterized by  $\theta_m$ . This function computes a similarity matrix  $\mathbf{S}$  between features of  $y$  and  $y_r$ :

$$S_{ij} = \frac{\exp(\langle \phi(y_i), \phi(y_{r,j}) \rangle / \tau)}{\sum_k \exp(\langle \phi(y_i), \phi(y_{r,k}) \rangle / \tau)}, \quad (4)$$

where  $\phi(\cdot)$  is a learnable feature transformation that maps input features to a higher-dimensional space,  $\langle \cdot, \cdot \rangle$  denotes

inner product, and  $\tau$  is a temperature parameter. We also introduce a learnable alignment module within the CLM to refine the alignment between reference and target features:  $y_a = \mathcal{F}_a(y, y_m; \theta_a)$ , where  $\mathcal{F}_a$  is implemented as a series of deformable convolution layers operating at multiple scales.

**(2) Conditional Latent Synthesis.** In the final stage of our feature matching and alignment strategy, we develop a Conditional Latent Synthesis (CLS) module to fuse the aligned reference features with the target image feature. We model this fusion process as a conditional probability with learnable weights:

$$p(y_f | y, y_a) = \mathcal{N}(\mu(y, y_a), \sigma^2(y, y_a)), \quad (5)$$

where  $y_f$  is the final latent representation, and  $\mu(\cdot)$  and  $\sigma^2(\cdot)$  are learnable functions implemented as neural networks. These functions estimate the mean and variance of the Gaussian distribution for  $y_f$  conditioned on both  $y$  and  $y_a$ . The mean function  $\mu(\cdot)$  is designed to incorporate adaptive weighting:

$$\mu(y, y_a) = \alpha \odot y + (1 - \alpha) \odot y_a, \quad (6)$$

where  $\alpha$  are dynamically computed weights based on content:  $\alpha = \sigma(\mathcal{F}_w([y, y_a]; \theta_f))$ . Here,  $\sigma$  is the sigmoid function, and  $\mathcal{F}_w$  is a small neural network predicting optimal fusion weights. This conditional generation approach with adaptive weights allows our model to capture complex dependencies between the input image and the reference image from the dictionary in the latent space, resulting in more flexible and powerful conditional coding. During training, we sample from this distribution to obtain  $y_f$ , while during inference, we use the mean  $\mu(y, y_a)$  as the final latent representation. This probabilistic formulation enables our model to handle uncertainties in the feature integration process and potentially generate diverse latent representations during training, which can improve the robustness and generalization capability of our deep compression system.

**(3) Entropy Coding and Hyperprior.** To further improve compression efficiency, we introduce a hyperprior network  $h_a$  that estimates a hyperprior  $z$  from the conditional latent  $y_f = h_a(y_f)$ . This hyperprior  $z$  provides additional context for more accurate probability estimation of  $y_f$ , enhancing the entropy model. The hyperprior is quantized and encoded separately,  $\hat{z} = Q(z)$ , where  $Q(\cdot)$  denotes the quantization operation.

For entropy coding, we adopt a slice-based autoregressive context model (). The conditional representation  $y_f$  is divided into  $K$  slices:  $y_f = [y_f^1, y_f^2, \dots, y_f^K]$ . The probability distribution of each slice is estimated using both previously processed slices and the hyperprior information. For the  $i$ -th slice, the probability model is expressed as:

$$p(y_f^i | y_f^{<i}, \hat{z}) = f_\theta(y_f^{<i}, \hat{z}), \quad (7)$$

where  $f_\theta$  is a neural network parameterized by  $\theta$ , and  $y_f^{<i} = [y_f^1, \dots, y_f^{i-1}]$  represents all previously encoded slices. The output of  $f_\theta$  is used to parametrize a probability distribution. Specifically, we model each element of  $y_f^i$  as a Gaussian distribution with mean  $\mu_i$  and scale  $\sigma_i$ :

$$p(y_f^i | y_f^{<i}, \hat{z}) \sim \mathcal{N}(\mu_i, \sigma_i^2), \quad (8)$$

where  $\Phi_i = (\mu_i, \sigma_i) = f_\theta(y_f^{<i}, \hat{z})$ . Here,  $\Phi_i$  represents the distribution parameters for the  $i$ -th slice. This approach captures complex dependencies within the latent representation, leading to more efficient compression. During the entropy coding process, we compute a residual  $r_i$  for each slice:  $r_i = y_f^i - \hat{y}_f^i$ , where  $\hat{y}_f^i$  is the quantized version of  $y_f^i$ . This residual helps to reduce quantization errors and improve reconstruction quality. The actual encoding process involves quantizing  $y_f^i - \mu_i$  and entropy encoding the result using the estimated distribution  $\mathcal{N}(0, \sigma_i^2)$ . During decoding, we reconstruct  $\hat{y}_f^i$  as  $\hat{y}_f^i = Q(y_f^i - \mu_i) + \mu_i$ , where  $Q(\cdot)$  denotes the quantization operation.

**(4) Decoding and Optimization.** During decoding, we first reconstruct  $\hat{z}$  and  $\hat{y}_f$  from the bitstream. Then, using the dictionary indices passed from the encoder, we apply the same reference processing and alignment procedure to reconstruct  $y$  from  $\hat{y}_f$ . Next,  $y$  is fed into the synthesis transform  $g_s$  to produce the final reconstructed image  $\hat{x}$ . It is important that we employ the same conditional latent synthesis pipeline on the decoder side to ensure consistency. The combination of the hyperprior  $z$  and the slice-based autoregressive model enables our system to achieve a fine balance between capturing global image statistics and local, contextual information, resulting in improved compression performance. To optimize our network end-to-end, we minimize the rate-distortion function:

$$L = D(x, \hat{x}) + \lambda R(b), \quad (9)$$

where  $D(x, \hat{x})$  is the distortion between the original and reconstructed images,  $R(b)$  is the bitrate of the encoded stream, and  $\lambda$  is an adaptive coefficient used to balance the rate-distortion trade-off. This optimization balances compression efficiency and reconstruction quality, allowing our approach to effectively leverage the aligned reference information at both the encoder and decoder stages.

### 3.4 Theoretical Perturbation Analysis

In image compression with auxiliary information, some degree of error in feature retrieval is inevitable due to the inherent complexity of the problem and the presence of noise. Understanding the bounds of this error is crucial for assessing and improving compression algorithms. We present a theoretical framework that quantifies these errors and provides insights into the factors affecting compression performance.

We formulate the problem as a rate-distortion optimization:

$$\min_{G_1, G_2, D} \mathbb{E}[R(G_1(x), G_2(\tilde{x})) + \lambda \mathcal{D}(x, D(G_1(x), G_2(\tilde{x})))]$$

where  $x \in \mathbb{R}^d$  is the original image,  $\tilde{x} \in \mathbb{R}^d$  the auxiliary image,  $G_1$  and  $G_2$  are encoders,  $D$  is a decoder,  $R$  is the rate loss, and  $\mathcal{D}$  is the distortion loss.

Our analysis is based on several key assumptions. We model the original image using a spiked covariance model:  $x = U^s + \xi$ , and the auxiliary image similarly:  $\tilde{x} = U^* \tilde{s} + \tilde{\xi}$ . The rate loss is entropy-based:  $R(z, \tilde{z}) = \mathbb{E}[-\log_2 p_\theta(z|\tilde{z})]$ , while the distortion loss is mean squared error:  $\mathcal{D}(x, \hat{x}) = \|x - \hat{x}\|^2$ . We assume sub-Gaussian noise with parameter  $\sigma^2$ ,

and allow for possible irrelevant information in the auxiliary image, with proportion  $p \in [0, 1)$ .

Our theoretical analysis aims to quantify the error in feature retrieval when using auxiliary information for image compression, specifically establishing an upper bound on the error in estimating the feature subspace of the original image, with a focus on the impact of irrelevant information in the auxiliary image. This analysis provides a rigorous foundation for understanding our Conditional Latent Coding (CLC) method, quantifies trade-offs between factors affecting compression performance, and offers insights into the method's robustness to imperfect auxiliary data. By emphasizing the importance of minimizing irrelevant information, it guides the design and optimization of our dictionary construction process. By deriving this error bound, we bridge the gap between theoretical understanding and practical implementation, providing a solid basis for the development and refinement of our compression algorithm.

Our main result quantifies the unavoidable error in feature retrieval:

**Theorem 1.** For any  $\delta > 0$ , with probability at least  $1 - \delta$ :

$$\|\sin \Theta(\Pr(\hat{G}_1), U^*)\|_F \leq C \left( \sqrt{r} \wedge \sqrt{\frac{r}{1-p}} \sqrt{\frac{(r+r(\Sigma_\xi)) \log(d/\delta)}{n}} \right)$$

where  $C > 0$  is a constant,  $p$  is the proportion of irrelevant parts in the auxiliary image,  $n$  is the number of training samples,  $r(\Sigma_\xi)$  is the effective rank of the noise covariance matrix, and  $\hat{G}_1$  is the estimated encoder for the original image.

This bound provides key insights: it reveals a trade-off between problem dimensionality ( $r$ ), sample size ( $n$ ), noise structure ( $r(\Sigma_\xi)$ ), and auxiliary image quality ( $p$ ). The system's tolerance to irrelevant information is quantified by  $\frac{1}{1-p}$ , while noise complexity is captured by the effective rank  $r(\Sigma_\xi)$ . The result also suggests potential for mitigation through increased sample size or improved auxiliary image quality.

## 4 Experimental Results

In this section, we provide extensive experimental results to evaluate the proposed CLC method and ablation studies to understand its performance.

### 4.1 Experimental Settings

**(1) Datasets.** In our experiments, we use two benchmark datasets: Flickr2W (Liu et al. 2020) and Flickr2K (Timofte et al. 2017). The Flickr2W dataset, containing 20,745 high-quality images, was used for training our model. To construct the image feature dictionary, we employed the Flickr2K dataset, which comprises 2,650 images. These Flickr2K images were randomly cropped into 256x256 patches to build the feature reference dictionary. We evaluated our algorithm on the Kodak (Kodak 1993) and CLIC (Toderici et al. 2020) datasets to evaluate its performance.

**(2) Implementation Details.** Our model was implemented using PyTorch and trained on 8 NVIDIA RTX 3090 GPUs. We trained the network for 30 epochs using the Adam

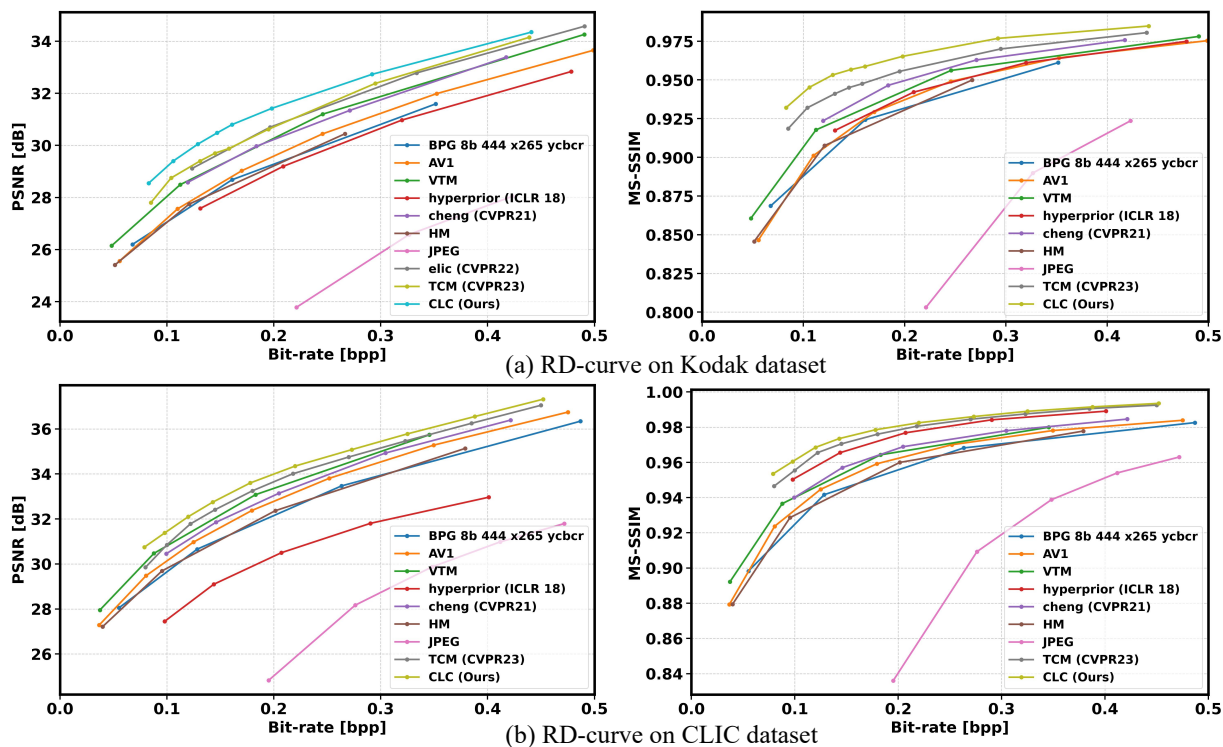


Figure 4: The rate-distortion performance comparison of different methods.

optimizer with an initial learning rate of  $1 \times 10^{-4}$ , which was reduced by a factor of 0.5 every 10 epochs. The batch size was set to 16 for each GPU. For the patch matching module, we used a patch size of  $16 \times 16$  pixels. The initial value of the adaptive fusion weight  $\alpha$  was set to 0.5. The number of slices  $K$  in the slice-based autoregressive context model was set to 8. In the KV-cache, we set the dimension of keys  $d_k$  and values  $d_v$  to 256. The cache size  $N$  was initially set to 300 and dynamically adjusted based on the GPU memory availability. The number of clusters  $K$  in Mini-Batch K-means was set to 3,000.

## 4.2 Performance Results

We report the rate-distortion results in Figure 4, showing our proposed CLC method outperforms existing methods across different bit-rates. The compared methods include traditional codecs like BPG (Bellard 2014), VTM (Bross et al. 2021), HM (Sullivan et al. 2012), and JPEG (Wallace 1992), as well as recent learning-based methods: the hyperprior model (Ballé et al. 2018), Cheng et al.’s approach (Cheng et al. 2021), ELIC (Zou et al. 2022), and TCM (Chen et al. 2023). We also include results from AV1 (Chen et al. 2018) for comparison. The improvement in compression efficiency is significant. On Kodak at MS-SSIM 0.95, CLC achieves 0.1 bpp, while TCM, VTM, BPG, and JPEG require 0.15, 0.18, 0.22, and 0.38 bpp, respectively, representing a 1.5 to 3.8 times increase in compression ratio. On CLIC at 34 dB PSNR, CLC achieves 0.2 bpp, compared to 0.25, 0.28, 0.35, and 0.45 bpp for TCM, VTM, Hyperprior, and JPEG, indicating larger efficiency gains. Figure 5 demonstrates our method’s superior performance in preserving detailed tex-

tures, particularly horizontal and vertical structures at low bit rates, as seen in railings and architectural features.

## 4.3 Ablation Studies

We conducted ablation studies to evaluate components of our CLC method, focusing on reference images, dictionary cluster size, and component contributions. We report results on both Kodak and CLIC datasets to demonstrate the performance across different image types.

**(1) Ablation Studies on the Number of Reference Images.** We changed the number of reference images from 1 to 5 to examine the impact on compression performance. Table 1 shows BD-rate savings compared to the VTM method with different numbers of reference images.  $\text{BD-Rate}_P$  represents savings in PSNR, while  $\text{BD-Rate}_M$  represents savings in MS-SSIM. Using three reference images achieves the best performance on both datasets, saving 14.5% and 13.9% BD-rate on Kodak and CLIC, respectively. More than three images introduce redundancy, degrading performance.

**(2) Ablation Studies on the Dictionary Cluster Size.** We conducted experiments with different dictionary cluster sizes to find the balance between compression efficiency and computational complexity. Table 2 shows the BD-rate savings and encoding time for different cluster sizes. A cluster size of 3000 provides the best trade-off between performance and complexity for both datasets, achieving significant BD-rate savings with reasonable encoding times. The sharp increase in the encoding time for cluster sizes beyond 3000 highlights the importance of carefully selecting this

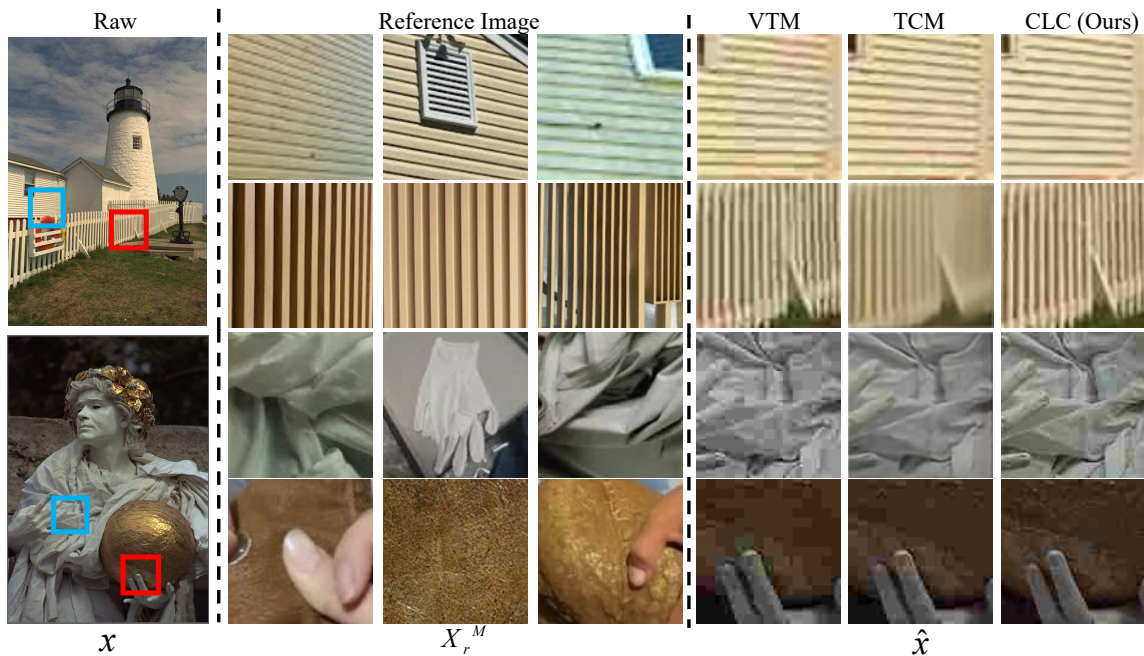


Figure 5: Image reconstruction results at around 0.1 bpp. From left to right: Raw inputs, reference images, reconstructed images. Red and blue boxes highlight specific areas of improvement.

Num of Ref. Images	Kodak		CLIC	
	BD-Rate <sub>P</sub>	BD-Rate <sub>M</sub>	BD-Rate <sub>P</sub>	BD-Rate <sub>M</sub>
1	-10.2	-11.5	-9.8	-10.9
2	-12.8	-13.7	-12.1	-13.2
3	<b>-14.5</b>	<b>-15.2</b>	<b>-13.9</b>	<b>-14.7</b>
4	-14.3	-15.0	-13.7	-14.5
5	-14.2	-14.9	-13.6	-14.4

Table 1: BD-rate savings (%) vs. VTM for different numbers of reference images.

Cluster Size	Kodak		CLIC		Encoding Time (s)
	BD-Rate <sub>P</sub>	BD-Rate <sub>M</sub>	BD-Rate <sub>P</sub>	BD-Rate <sub>M</sub>	
1000	-11.8	-12.5	-11.2	-11.9	0.52
2000	-13.7	-14.3	-13.1	-13.8	0.78
3000	<b>-14.5</b>	<b>-15.2</b>	<b>-13.9</b>	<b>-14.7</b>	1.05
4000	-14.6	-15.3	-14.0	-14.8	2.31
5000	-14.7	-15.4	-14.1	-14.9	5.67

Table 2: BD-rate savings (%) vs. VTM and encoding time for different dictionary cluster sizes

parameter to balance compression efficiency and computational cost.

### (3) Ablation Studies on Major Algorithm Components.

We conducted ablation experiments to evaluate the contribution of major components. Table 3 shows each component’s impact on the Kodak dataset performance. All components contribute significantly, with CLS having the most substantial impact (4.7% BD-rate savings), highlighting the importance of adaptive feature modulation. The

Model Configuration	BD-Rate Savings		Encoding Time (s)
	BD-Rate <sub>P</sub>	BD-Rate <sub>M</sub>	
Full Model	<b>-14.5</b>	<b>-15.2</b>	1.05
w/o CLM	-12.3	-13.1	0.98
w/o CLS	-9.8	-10.5	0.92
w/o KV-cache	-14.4	-15.1	1.87
w/o Multi-example Query	-13.8	-14.6	0.97

Table 3: BD-rate savings (%) vs. VTM for different model configurations on Kodak dataset

KV-cache, while minimally impacting compression performance, significantly reduces encoding time (from 1.87s to 1.05s). Multi-sample query in dictionary construction improves BD-rate savings by 0.7% (BD-Rate<sub>P</sub>) and 0.6% (BD-Rate<sub>M</sub>), enhancing overall compression capability through more diverse representations.

## 5 Conclusion

This study proposes Conditional Latent Coding (CLC), a novel deep learning-based image compression method that dynamically generates latent reference representations through a universal image feature dictionary. We develop innovative techniques for dictionary construction, efficient search/matching, alignment, and fusion, with theoretical analysis of robustness to dictionary and latent perturbations. While focused on compression, CLC’s adaptive feature utilization principles may inspire broader vision tasks. Future work includes balancing compression efficiency and visual information utilization to address growing data transmission demands.

## Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 62331014) and Grant 2021JC02X103.

## References

- Ayzik, S.; and Avidan, S. 2020. Deep image compression using decoder side information. In *ECCV*, 699–714.
- Ballé, J.; Laparra, V.; and Simoncelli, E. P. 2017. End-to-end optimized image compression. In *ICLR*.
- Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. In *ICLR*.
- Bellard, F. 2014. BPG Image Format. <https://bellard.org/bpg/>.
- Bross, B.; Chen, J.; Ohm, J.-R.; Sullivan, G. J.; and Wang, Y.-K. 2021. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Chen, Y.; Mukherjee, D.; Han, J.; Grange, A.; Xu, Y.; Liu, Z.; Parker, S.; Chen, C.; Agarwal, H.; Deshpande, S.; et al. 2018. An Overview of Core Coding Tools in the AV1 Video Codec. In *PCS*. IEEE.
- Chen, Z.; Wang, R.; He, D.; Zhang, L.; and Ma, S. 2023. Transformer-based Context Modeling for Image Compression. In *CVPR*.
- Cheng, Z.; Sun, H.; Takeuchi, M.; and Katto, J. 2020. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *CVPR*, 7939–7948.
- Cheng, Z.; Sun, H.; Takeuchi, M.; and Katto, J. 2021. Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules. In *CVPR*, 7939–7948.
- Huang, Y.; Chen, B.; Qin, S.; Li, J.; Wang, Y.; Dai, T.; and Xia, S.-T. 2023. Learned distributed image compression with multi-scale patch matching in feature domain. In *AAAI*, volume 37, 4322–4329.
- Jia, Z.; Li, J.; Li, B.; Li, H.; and Lu, Y. 2024. Generative Latent Coding for Ultra-Low Bitrate Image Compression. In *CVPR*, 26088–26098.
- Kodak, E. 1993. Kodak Lossless True Color Image Suite (PhotoCD PCD0992). Version 5.
- Lee, J.; Cho, S.; and Beack, S.-K. 2019. Context-adaptive Entropy Model for End-to-end Optimized Image Compression. In *ICLR*.
- Li, J.; Li, B.; and Lu, Y. 2021. Deep contextual video compression. In *NeurIPS*, volume 34, 18114–18125.
- Li, J.; Li, B.; and Lu, Y. 2023. Neural video compression with diverse contexts. In *CVPR*, 22616–22626.
- Li, M.; Shen, L.; Ye, P.; Feng, G.; and Wang, Z. 2023. RFD-ECNet: Extreme Underwater Image Compression with Reference to Feature Dictionary. In *ICCV*, 12980–12989.
- Liu, J.; Lu, G.; Hu, Z.; and Xu, D. 2020. A unified end-to-end framework for efficient deep image compression. *arXiv preprint arXiv:2002.03370*.
- Liu, J.; Sun, H.; and Katto, J. 2023. Learned image compression with mixed transformer-cnn architectures. In *CVPR*, 14388–14397.
- Mentzer, F.; Agustsson, E.; Tschannen, M.; Timofte, R.; and Van Gool, L. 2018. Conditional probability models for deep image compression. In *CVPR*, 4394–4402.
- Minnen, D.; Ballé, J.; and Toderici, G. D. 2018. Joint autoregressive and hierarchical priors for learned image compression. In *NeurIPS*, 10771–10780.
- Qian, Y.; Lin, M.; Sun, X.; Tan, Z.; and Jin, R. 2022. Entroformer: A transformer-based entropy model for learned image compression. *arXiv preprint arXiv:2202.05492*.
- Sheng, X.; Li, J.; Li, B.; Li, L.; Liu, D.; and Lu, Y. 2022. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*, 25: 7311–7322.
- Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; and Wiegand, T. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12): 1649–1668.
- Sun, J.; Zheng, N.-N.; Tao, H.; and Shun, H.-Y. 2003. Image hallucination with primal sketch priors. In *CVPR*.
- Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.-H.; and Zhang, L. 2017. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPR workshops*, 114–125.
- Toderici, G.; Shi, W.; Timofte, R.; Theis, L.; Ballé, J.; Agustsson, E.; Johnston, N.; and Mentzer, F. 2020. CLIC: Workshop and challenge on learned image compression. In *CVPR workshop*.
- Toderici, G.; Vincent, D.; Johnston, N.; Jin Hwang, S.; Minnen, D.; Shor, J.; and Covell, M. 2017. Full resolution image compression with recurrent neural networks. In *CVPR*, 5306–5314.
- Wallace, G. K. 1992. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1): xviii–xxxiv.
- Wu, Y.; Chen, Z.; Wen, Y.; Huang, H.; and Yuan, H. 2022. Content-aware reference frame selection for efficient video compression. *IEEE Transactions on Image Processing*, 31: 5186–5198.
- Xiong, Z.; Sun, X.; and Wu, F. 2010. Robust web image/video super-resolution. *IEEE transactions on image processing*, 19(8): 2017–2028.
- Yang, R. 2023. Tinc: Tree-structured implicit neural compression. In *CVPR*, 18517–18526.
- Yue, H.; Sun, X.; Yang, J.; and Wu, F. 2013. Cloud-based image coding for mobile devices—Toward thousands to one compression. *IEEE transactions on multimedia*, 15(4): 845–857.
- Zhao, J.; Li, B.; Li, J.; Xiong, R.; and Lu, Y. 2021. A universal encoder rate distortion optimization framework for learned compression. In *CVPR*, 1880–1884.
- Zhao, J.; Li, B.; Li, J.; Xiong, R.; and Lu, Y. 2023. A universal optimization framework for learning-based image codec. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(1): 1–19.

Zou, F.; Feng, Y.; Wei, Y.; and Ren, J. 2022. ELIC: Efficient Learned Image Compression with Unevenly Grouped Space-Channel Contextual Adaptive Coding. In *CVPR*, 5718–5727.