IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 November 2024, accepted 17 December 2024, date of publication 25 December 2024,
date of current version 31 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3522238

RESEARCH ARTICLE

Dual-Conditioned Training to Exploit Pre-Trained Codebook-Based Generative Model in Image Compression

SHOMA IWAI, (Graduate Student Member, IEEE), TOMO MIYAZAKI, (Member, IEEE), AND SHINICHIRO OMACHI, (Senior Member, IEEE)
Graduate School of Engineering, Tohoku University, Sendai, Miyagi 980-8579, Japan

Corresponding author: Shoma Iwai (shoma.iwai.s4@dc.tohoku.ac.jp)

This work was supported in part by Japan Society for the Promotion of Science under Grant JP24KJ0343.

ABSTRACT Learned image compression (LIC) is increasingly gaining attention. To improve the perceptual quality of reconstructions, generative LIC has been studied, using generative models such as Generative Adversarial Networks (GANs). State-of-the-art generative LIC methods have achieved remarkable performance even in low bit rate settings. Unlike most approaches trained from scratch, we propose a generative LIC that utilizes a pre-trained codebook-based generative model, Vector-Quantized GAN (VQGAN). Specifically, our model is designed to exploit its powerful image-generation capabilities to enhance compression performance. Our approach reconstructs an image from a transmitted bitstream in two steps: (1) estimating VQGAN tokens and feeding them into the pre-trained VQGAN decoder, and (2) modifying the decoder's intermediate features to address artifacts and distortions. Our preliminary experiments reveal that the information allocation between (1) and (2) is pivotal for reconstruction quality. Moreover, we found that the ideal allocation varies based on the target bit rate. Motivated by these findings, we propose a novel Dual-Conditioned training. Through the training, the model learns to adjust the total bit rate and information allocation between (1) and (2) based on two conditional inputs. Subsequently, we explore the conditional inputs to achieve the optimal results for each target bit rate. This training strategy enables us to effectively exploit the generation capability of VQGAN across different bit rates. Our method, named Dual Conditioned VQGAN-based Image Compression (DC-VIC), outperforms state-of-the-art generative LIC methods in rate-distortion-perception performance. Code will be available at https://github.com/iwa-shi/DC_VIC

INDEX TERMS Generative adversarial networks, image compression, VQGAN.

I. INTRODUCTION

Image compression is a pivotal technology in the digital era. While traditional image compression algorithms (e.g., JPEG, HEVC, and VVC) are hand-crafted, learned image compression (LIC) has been studied for several years [1], [2], [3], [4]. Utilizing end-to-end optimization using large image datasets, state-of-the-art LIC methods [5], [6] have surpassed the performance of the latest standard codec, VVC [7]. However, most LIC methods optimized for rate-distortion (RD) performance suffer from blurred artifacts in reconstructions at lower bit rates. To address this, several methods [8], [9], [10], [11], [12], [13] aim to enhance rate-distortion-perception (RDP) performance [14]. Such methods use generative models (e.g., Generative Adversarial Networks (GANs) [15] and Diffusion model [16]), significantly improving the perceptual quality of reconstructed images. We refer to these methods as *generative LIC*.

While most generative LIC models are trained from scratch, the integration of pre-trained models into generative LIC has been underexplored. In this paper, we leverage a pre-trained Vector Quantized GAN (VQGAN) [17] for

The associate editor coordinating the review of this manuscript and approving it for publication was Yun Zhang.

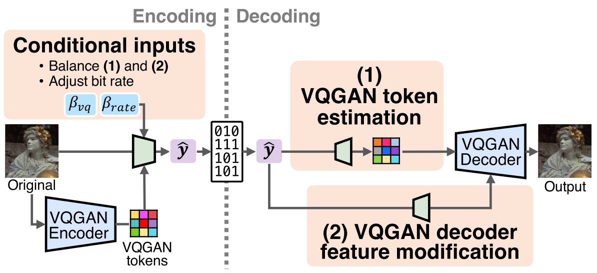


FIGURE 1. **High-level overview of our DC-VIC. It reconstructs the original image by two processes: (1) VQGAN token estimation and (2) VQGAN decoder feature modification. Accurate token estimation is important to leveraging the VQGAN decoder's image generation ability. On the other hand, the decoder feature modification improves reconstruction fidelity by rectifying artifacts and distortion introduced by the VQGAN decoder. To adjust the total bit rate and information allocation between (1) and (2) in the encoding part, we introduce conditional inputs, β_{rate} and β_{vq}, respectively. We adjust these two inputs to obtain high compression performance for various target rates.**

generative LIC. VQGAN uses an encoder to convert an image into a sequence of discrete tokens through vector quantization with a learnable codebook. The decoder then reconstructs the image from these tokens. The main reasons for incorporating VQGAN into our LIC model are twofold. Firstly, the codebook's capacity to represent a wide range of images is valuable. A compact yet expressive codebook facilitates effective image compression, akin to the concept of traditional dictionary-based compression methods [18], [19], [20]. Secondly, the decoder's capability to reconstruct high-quality images has been demonstrated in various image generation applications [21], [22], [23]. Motivated by these attributes, we aim to fully exploit the potential of pre-trained VQGAN for enhanced image compression.

The high-level overview of our Dual-Conditioned VQGAN-based Image Compression (**DC-VIC**) is illustrated in Fig. 1. On the encoder side, the original image and its VQGAN tokens are merged and encoded into a single bitstream. The decoding process involves two key steps: (1) VQGAN token estimation and (2) modification of the VQGAN decoder's features. Initially, VQGAN tokens are estimated and fed into the pre-trained VQGAN decoder for image reconstruction. During the reconstruction, we modify the decoder's intermediate features to mitigate artifacts and distortions introduced by VQGAN. This adjustment significantly enhances reconstruction fidelity compared to using VQGAN reconstruction directly as the final output [24], [25]. Accurate token estimation is pivotal for exploiting the VQGAN decoder's image reconstruction capabilities. On the other hand, modifying the decoder's features is also important to enhance reconstruction fidelity. However, the amount of information sent to the decoder side is limited, which prompts the question: *How should we determine the allocation of information between (1) token estimation and (2) feature modification?* Our preliminary experiments revealed that this information allocation is crucial for enhancing RDP performance. In addition, we found that the optimal allocation varies according to the target bit rate.

A straightforward approach for finding the optimal allocation would be training separate LIC models with various hyperparameters. However, this leads to substantial training costs. To address this issue and efficiently balance information allocation, we introduce the novel **Dual Conditioned Training**. Our LIC model incorporates two conditional inputs, β_{rate} and β_{vq}, as illustrated in Fig. 1. These inputs are integrated into the model, enabling it to dynamically adjust the overall bit rate and information allocation between token estimation and VQGAN decoder feature modification within a single model. After the training, we explore the conditional inputs to achieve optimal compression performance for each target bit rate. Moreover, we fine-tune the model using only the selected conditional inputs, further improving the performance. This training strategy enables our DC-VIC to effectively exploit the generation capabilities of pre-trained VQGAN at different bit rates without training multiple models.

Our contributions are summarized as follows:

- We propose a novel Dual-Conditioned VQGAN-based Image Compression (DC-VIC). Unlike existing methods [24], [25], it reconstructs images via (1) VQGAN token estimation and (2) VQGAN decoder feature modification.
- Our preliminary experiments demonstrate that the balance of information allocation between token estimation and decoder feature modification is critical for achieving optimal compression performance. Furthermore, the optimal allocation varies depending on the target bit rates. To the best of our knowledge, this is the first work that has found these phenomena.
- To effectively adjust the information allocation and total bit rate within a single model, we propose Dual Conditioned Training. It enables us to explore optimal conditional inputs for each target bit rate. We also conducted extensive experiments to validate each strategy within our training method.
- DC-VIC outperforms the state-of-the-art generative LIC methods in RDP performance while using a single model for multiple bit rates. Our method also demonstrates robust performance across various VQGAN variations.

II. RELATED WORK

A. LEARNED IMAGE COMPRESSION (LIC)

Since the first end-to-end VAE-based learned image compression method [1], a number of methods have been proposed. Typical LIC models are optimized to minimize the following rate-distortion (RD) loss function:

$$\mathcal{L} = \lambda R + D, \tag{1}$$

where R represents the approximated bit rate after compression and D denotes the distortion measured by metrics such as MSE and MS-SSIM. A hyperparameter λ determines the balance between bit rate and distortion. To improve RD performance, some studies have focused on enhancing entropy estimation, including Hyperprior [2], 2D context model [26],

[27], Channel Autoregressive Model (Charm) [3], M2T [6], Multirate Progressive Entropy Model [28], Transformer-based Entropy Model [29], Multi-Reference Entropy Model [30] and Checkerboard Entropy Model [31]. Other studies have improved the architecture of encoders and decoders. Specifically, several approaches have incorporated advanced modules such as Residual Blocks [32], Multi-scale Residual Blocks [33], Lightweight Attention modules [34], Graph-attention [35] and Shifted Window-based Attention [5], [36], [37]. Additionally, another line of research explores overfitting-based LIC methods [38], [39], [40], which optimize lightweight neural networks for single images and transmit the network weights. These advanced entropy estimators and modern deep-learning architectures have significantly improved rate-distortion performance, surpassing even the latest standard codec, VVC [7].

B. GENERATIVE LIC

While LIC methods achieved superior RD performance, preserving perceptual quality at a low bit rate is still challenging. Due to the distortion-perception trade-off [41], LIC models optimized to minimize distortion suffer from reduced perceptual quality. Specifically, these models tend to reconstruct images with blur artifacts in an effort to minimize average distortion (e.g., pixel-level squared error). This issue becomes more pronounced at low bit rates, where there is insufficient information to accurately reconstruct fine details. To mitigate this problem, generative LIC methods have been studied [8], [9], [10], [11], [12], [42], [43], [44], [45], [46]. These methods incorporate generative models such as generative adversarial networks (GAN) [15] and diffusion model [16], improving rate-distortion-perception (RDP) trade-off [14].

While most existing generative LIC methods are trained from scratch, we propose a novel generative LIC model that uses a powerful pre-trained generative model, Vector Quantized GAN (VQGAN) [17]. Only a few existing works have utilized a pre-trained model in generative LIC. Xu et al. [43] proposed a LIC method using the inversion of a pre-trained diffusion model to achieve high perceptual quality. However, due to its expensive computation cost, the application is limited to low-resolution images. ILLM [12] uses a pre-trained VQ-VAE [47] to train the discriminator. Hence, the image-generation ability of VQ-VAE is not directly utilized for image reconstruction. Mao et al. [24] and Jiang et al. [25] leverage pre-trained VQGAN autoencoder for image reconstruction. Mao et al. [24] transform VQGAN token indices into a bitstream using a zip algorithm, with the bit rate controlled by dynamically adjusting the codebook size via K-means clustering. Jiang et al. [25] adjust the bit rate by masking and transmitting a subset of the token indices. However, both methods use the VQGAN's reconstruction as the final output image. This limits reconstruction fidelity due to the inherent constraints of the VQGAN autoencoder, resulting in higher distortion. In contrast, we modify the intermediate feature in the VQGAN decoder via spatial feature transform (SFT) layers [48], [49], improving reconstruction fidelity. Moreover, our Dual-Conditioned training optimizes the information balance between VQGAN token estimation and decoder feature modification, providing a key distinction from other approaches.

C. CONDITIONAL TRAINING IN LIC

Some LIC methods use conditional training to obtain an adaptive model. In conditional training, the LIC model takes a conditional input as well as an original image to control the model behavior. Most of these methods are designed for variable-rate LIC, where a single model can compress an image into different bit rates. Specifically, λ in (1) is used as a conditional input for the model. The model is then trained to adjust the bit rate according to the condition. Some methods [51], [52], [53] focus on image compression for machine vision tasks, where the LIC model is optimized to enhance the performance of downstream tasks, such as classification and segmentation, after compression. These models are conditionally trained to adapt to various tasks, enhancing performance across multiple applications. Additionally, other methods [9], [10] use conditional training to control distortion-perception trade-off [41]. It enables users to choose high-fidelity (less distortion) compression or high-realism compression, improving practicality.

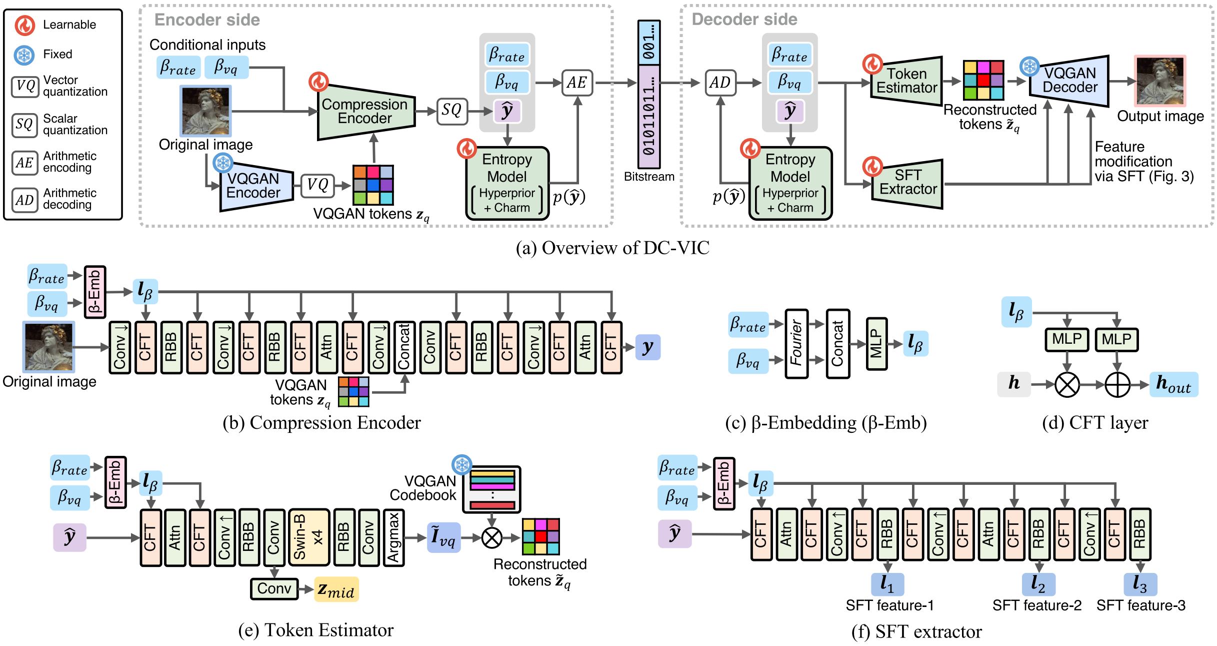
To integrate the conditional input into the model, several methods have been proposed, such as scaling latent representation [54], conditional convolution [55], interpolation channel attention [56], spatial feature transform [57], β-conditioning [9], and visual prompt [53]. We use a similar method as β-conditioning [9] to inject conditional inputs. However, instead of adjusting the distortion-perception trade-off, we use it to control two different factors: total bit rate and the information allocation between VQGAN token estimation and decoder feature modification. We will show the effectiveness of controlling these factors with a single model to achieve optimal RDP performance.

D. VQGAN AND ITS APPLICATION

In image generation research, vector quantization (VQ) is widely used. Guyon et al. introduced the Vector Quantized Variational AutoEncoder (VQ-VAE) [47], where each pixel in a latent representation is vector-quantized using a learned codebook. Specifically, an encoder E transforms an original image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times 3}$ into a latent $\hat{\boldsymbol{z}} = E(\boldsymbol{x}) \in \mathbb{R}^{h \times w \times n_z}$, where h, w, and n_z denote the height, width and number of channels, respectively. Each pixel in $\hat{\boldsymbol{z}}$ is replaced with the nearest token $z_k \in \mathbb{R}^{n_z}$ from the codebook $\mathcal{Z} = \{z_k\}_{k=1}^{K}$:

$$\boldsymbol{z}_q = \left(\underset{z_k \in \mathcal{Z}}{\arg\min} \, ||\hat{z}_{ij} - z_k|| \right) \in \mathbb{R}^{h \times w \times n_z}, \tag{2}$$

where K is the codebook size, and i, j represent the spatial position. The decoder G then reconstructs the image $\hat{\boldsymbol{x}} = G(\boldsymbol{z}_q)$. VQGAN [17] further extends this concept by integrating GAN-based training into VQ-VAE, improving the



(a) Overview of DC-VIC

(b) Compression Encoder

(c) β-Embedding (β-Emb)

(d) CFT layer

(e) Token Estimator

(f) SFT extractor

FIGURE 2. (a) The overview of the proposed DC-VIC and (b-f) the detailed architecture of each component. *VQ*, *SQ*, *AE*, and *AD* in (a) denote vector-quantization, scalar-quantization, arithmetic encoding, and arithmetic decoding, respectively. RBB in (b) is Residual Bottleneck Block [4]. *Fourier* in (c) represents the Fourier-encoding [50].

perceptual quality of reconstructions. Due to its powerful image reconstruction capability, VQGAN has been extensively employed in various image generation methods, such as MaskGIT [21], Token-Critic [22], VQ-Diffusion [23], and Latent Diffusion [58]. VQGAN-based autoencoder has also been utilized in image restoration studies. CodeFormer [49] incorporates a pre-trained VQGAN to realize high-fidelity face image restoration. It has introduced spatial feature transform (SFT) layers to improve reconstruction fidelity while fixing pre-trained VQGAN parameters. FeMaSR [59] introduced semantic regularization for training VQGAN, enhancing reconstruction quality.

While our decoding process is inspired by CodeFormer [49] and FeMaSR [59], it differs in the following aspects. Firstly, unlike the image restoration task, where only the degraded image is accessible, the original image is available to the encoder in image compression. Therefore, the encoder can selectively store essential information from the original image. We discovered that effective allocation of information on the encoder side is crucial for reconstruction performance. This phenomenon has never been discussed in existing studies and is unique to image compression tasks. Secondly, we propose dual-conditioned training to control the model behavior within a single model, dealing with the aforementioned challenge.

III. METHODOLOGY

In this section, we present our image compression model, termed Dual Conditioned VQGAN-based Image Compression (**DC-VIC**). We begin with an overview and detailed description of our compression model in Section III-A. Following this, Section III-B discusses preliminary experiments, which uncover the challenges that our dual-conditioned training strategy aims to overcome. Finally, we detail the three stages of our **Dual-Conditioned Training** in Section III-C, designed to address these challenges.

A. DC-VIC

1) OVERVIEW

Fig. 2(a) illustrates the overview of DC-VIC, which is comprised of six components: a compression encoder, entropy model, VQGAN encoder, token estimator, SFT extractor, and VQGAN decoder. We employ pre-trained VQGAN encoder and decoder, whose parameters are fixed. To facilitate our dual-conditioned training approach, we incorporate two additional conditional inputs, β_{rate} and β_{vq}, into the model and the loss function. β_{rate} adjusts the overall bit rate, while β_{vq} controls the accuracy of VQGAN token estimation. In the sections below, we provide a detailed explanation of the encoding and decoding processes within our model, as well as the discriminator and loss function used during training.

2) ENCODING PART

Our encoding process is designed to encode the information of the original image and its VQGAN token sequence into a single bitstream. Given an original image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$, the pre-trained VQGAN encoder extracts a latent code $\hat{z}$. It is then vector quantized using a pre-trained VQGAN codebook, obtaining the token sequence z_q and the indices $\boldsymbol{I}_{vq}$ of the selected token, where $I_{vq}^{(i)} \in \{0, 1, \cdots, K - 1\}$ and K denotes a codebook size. The compression

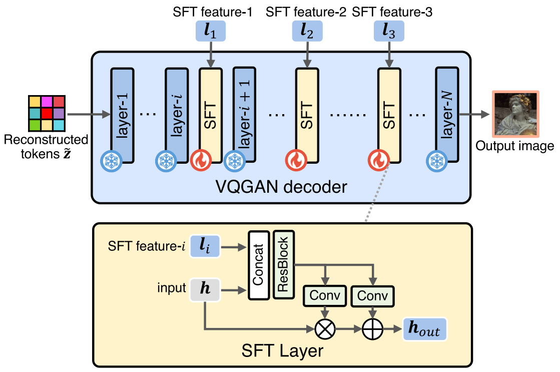


FIGURE 3. Intermediate feature modification via spatial feature transform (SFT) in VQGAN decoder. The parameters of the VQGAN decoder are fixed, while those of the SFT layers are trained.

encoder takes $\boldsymbol{x}$, z_q, and two conditional inputs β_{rate}, β_{vq} as inputs. The conditional inputs β_{vq}, β_{rate} are scalar values and pivotal to our dual-conditioned training, which will be detailed in Sec. III-C. These inputs are transformed by the encoder into a latent code $\boldsymbol{y}$. Fig. 2(b) depicts the encoder's architecture, which is inspired by ELIC [4] encoder and has multiple residual bottleneck blocks (RBB). To accommodate additional inputs z_q, β_{vq}, and β_{rate}, we add two modifications. First, the intermediate feature of the encoder is concatenated with z_q, allowing the encoder to integrate the features of the original image with the VQGAN tokens. Second, we introduce channel-wise feature transformations (CFT). In CFT, we first encode β_{vq}, β_{rate} into vectors using Fourier encoding $\gamma(\cdot)$ [50]. These vectors are then concatenated and converted into a single feature vector $\boldsymbol{l}_\beta$ with a multi-layer perception (MLP) as shown in Fig. 2(c):

$$\boldsymbol{l}_\beta = \text{MLP}\left(\text{concat}\left(\gamma(\beta_{vq}), \gamma(\beta_{rate})\right)\right) \tag{3}$$

$$\gamma(x) = \Big(\sin(2^0\pi x), \cos(2^0\pi x), \cdots, \sin(2^{L-1}\pi x), \cos(2^{L-1}\pi x)\Big), \tag{4}$$

where we set $L = 10$. Subsequently, an intermediate encoder feature $\boldsymbol{h}$ is transformed as follows, as illustrated in Fig. 2(d):

$$\boldsymbol{h}_{out} = \text{MLP}_{\text{scale}}(\boldsymbol{l}_\beta) \cdot \boldsymbol{h} + \text{MLP}_{\text{shift}}(\boldsymbol{l}_\beta). \tag{5}$$

These adaptations enable the compression encoder to effectively utilize the additional inputs. The output latent $\boldsymbol{y}$ is scalar-quantized to $\hat{\boldsymbol{y}}$, akin to existing image compression methods [3], [4]. The quantized latent $\hat{\boldsymbol{y}}$ is then converted into a bitstream through entropy coding. We estimate the entropy of $\hat{\boldsymbol{y}}$ using a hyperprior [2] and a channel-wise autoregressive model (Charm) [3]. The probability distribution $p(\hat{\boldsymbol{y}})$ is modeled as a Gaussian distribution, with the hyperprior and Charm determining its mean and scale parameters. Please refer to [3] for a more detailed procedure. We also encode β_{vq} and β_{rate} into the bitstream to use them in the decoding process, which takes less than one byte for each.

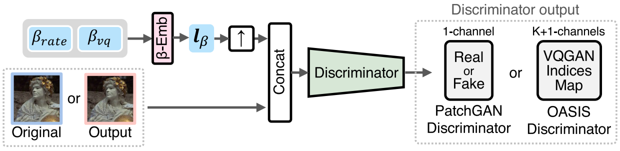


FIGURE 4. For the discriminator, we consider two options: PatchGAN [60] and OASIS [61]. While the PatchGAN discriminator predicts if the input image is real or fake, the OASIS discriminator predicts the VQGAN indices map, akin to performing semantic segmentation. We use β_{rate} and β_{vq} as conditional input for the discriminator.

3) DECODING PART

On the decoder side, the bitstream is first decoded back into $\hat{\boldsymbol{y}}$, β_{rate}, and β_{vq} using the same hyperprior and Charm as in the encoding process. We reconstruct an image through two main processes: (1) estimating VQGAN tokens z_q and (2) modifying intermediate features of the VQGAN decoder with spatial feature transform (SFT) layers [48], [59]. The token estimator, which consists of CNN blocks and Swin-Transformer-based [36] blocks, processes $\hat{\boldsymbol{y}}$, β_{rate}, and β_{vq} to predict the vector quantization indices $\boldsymbol{I}_{vq}$, as shown in Fig. 2(e). β_{rate} and β_{vq} are injected via CFT layers as in the compression encoder. Using the predicted indices $\tilde{\boldsymbol{I}}_{vq}$ and the pre-trained VQGAN codebook, we obtain the reconstructed tokens $\tilde{z}_q$, which are then fed into the VQGAN decoder. Following this, the SFT extractor takes $\hat{\boldsymbol{y}}$, β_{rate}, and β_{vq} as inputs and extracts multi-scale SFT features $\boldsymbol{l} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3\}$, as shown in Fig. 2(f). These SFT features are used to modify the VQGAN decoder's intermediate features during the image reconstruction. As detailed in Fig. 3, SFT layers [48] are inserted into the VQGAN decoder. Each SFT layer modifies a decoder intermediate feature $\boldsymbol{h}_i$ with the corresponding SFT feature $\boldsymbol{l}_i$ as follows:

$$\boldsymbol{h}_i^{out} = g_{\text{scale}}(\boldsymbol{l}_i) \odot \boldsymbol{h}_i + g_{\text{shift}}(\boldsymbol{l}_i), \tag{6}$$

where $\odot$ denotes element-wise multiplication, and $g_{\text{scale}}, g_{\text{shift}}$ represent convolution layers. The resulting modified feature $\boldsymbol{h}_i^{out}$ is then input into the subsequent layer of the VQGAN decoder, continuing the reconstruction process. By employing SFT layers, we dynamically refine the VQGAN decoder's intermediate features, thus improving reconstruction fidelity without updating the VQGAN's parameters. Finally, we obtain the reconstructed image $\hat{\boldsymbol{x}}$ from the VQGAN decoder.

4) DISCRIMINATOR

For the discriminator in our GAN-based training, we consider two options: PatchGAN [60] and OASIS discriminator [61], as illustrated in Fig 4. PatchGAN discriminator, which consists of several convolutional layers, is a common choice in generative LIC studies [9], [10], [13]. On the other hand, the OASIS discriminator was originally designed for a semantic image synthesis task, which aims to generate realistic images from their semantic label maps. Unlike the PatchGAN discriminator, which classifies images as real or fake, the OASIS discriminator performs semantic

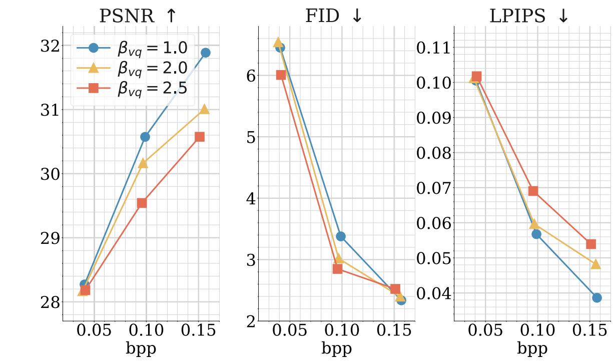


FIGURE 5. **The results of the preliminary experiments on CLIC2020 test dataset using nine distinct models trained with fixed (β_{vq}, β_{rate}) pairs.**

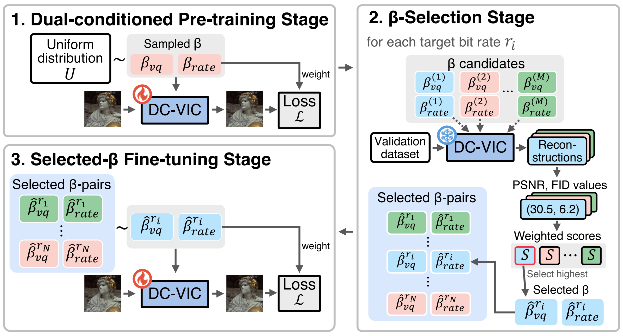


FIGURE 6. **The details of the dual-conditioned training. It includes three stages: dual-conditioned pre-training, β-selection, and selected-β fine-tuning.**

segmentation across $C+1$ classes, where C is the number of semantic classes, with an additional class for "fake" images. It is trained to identify the correct semantic label map for real images while assigning the fake label to all pixels in generated images. Recently, ILLM [12] incorporated this OASIS discriminator into generative LIC, leveraging VQ-VAE [47] codebook indices maps instead of semantic labels. While ILLM employed VQ-VAE solely to train the OASIS discriminator, our model already utilizes the indices map $\boldsymbol{I}_{vq}$ in the encoding process. It allows for seamless integration of the OASIS discriminator into our approach without extra model operations.

Also, as shown in Fig 4, we use β_{rate} and β_{vq} as conditional inputs for the discriminator. These conditional inputs are embedded using (3) and expanded to the size of the input image. Then, we concatenate the expanded $\boldsymbol{l}_{\beta}$ and image and feed it into the discriminator. Unless otherwise specified, we use the PatchGAN discriminator in our experiments for simplicity.

5) LOSS FUNCTION

Our compression model is trained with a combination of four loss functions: rate loss R, image loss $\mathcal{L}_{img}$, adversarial loss $\mathcal{L}_{adv}$, and VQGAN token reconstruction loss $\mathcal{L}_{vq}$, formulated as follows:

$$R = -\frac{1}{HW}\sum \log_2 p(\hat{y}) \tag{7}$$

$$\mathcal{L}_{img} = \lambda_{MSE}^{img}\text{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \text{LPIPS}(\boldsymbol{x}, \hat{\boldsymbol{x}}) \tag{8}$$

$$\mathcal{L}_{adv} = -\log D(\hat{\boldsymbol{x}}) \tag{9}$$

$$\mathcal{L}_{vq} = \text{CE}(\boldsymbol{I}_{vq}, \tilde{\boldsymbol{I}}_{vq}) + \lambda_{MSE}^{vq}\text{MSE}(z_q, z_{mid}), \tag{10}$$

where CE and D denote the cross-entropy loss and the discriminator, respectively. The rate loss R represents an approximated bit rate. It is computed using the probability distribution $p(\hat{y})$, estimated by the entropy model, as described in Sec. III-A2. The image loss is a weighted sum of the mean squared error (MSE) and the Learned Perceptual Image Patch Similarity (LPIPS) [62] between the original and reconstructed images. For the adversarial loss $\mathcal{L}_{adv}$, we apply the standard non-saturated adversarial loss. However, we use the cross-entropy adversarial loss instead of (9) for the OASIS discriminator [12]. These rate loss, image loss, and adversarial loss are commonly used in GAN-based image compression methods [9], [10], [13]. On the other hand, the VQGAN token reconstruction loss $\mathcal{L}_{vq}$ is specifically designed for precise estimation of VQGAN tokens. Following [59], $\mathcal{L}_{vq}$ includes two terms: (a) cross-entropy loss between the predicted and actual VQGAN codebook indices and (b) MSE loss between the projected intermediate feature z_{mid} in the token estimator and the vector-quantized VQGAN latent z_q. As shown in Fig. 2(e), we use an additional 1×1 convolution layer to transform the intermediate feature of the token estimator into z_{mid}, ensuring it shares the same dimensionality as z_q. The first term encourages accurate token index prediction, while the second term encourages the model to estimate tokens close to z_q.

The overall loss function $\mathcal{L}$ is a weighted sum of these individual losses:

$$\begin{aligned}\mathcal{L} = {} & \lambda_R \exp(\beta_{rate})R + \lambda_{img}\mathcal{L}_{img} \\ & + \lambda_{vq}\exp(\beta_{vq})\mathcal{L}_{vq} + \lambda_{adv}\mathcal{L}_{adv},\end{aligned} \tag{11}$$

where the conditional inputs, β_{rate} and β_{vq}, modulate the weights of R and $\mathcal{L}_{vq}$ exponentially, respectively. β_{rate} influences the overall bit rate, while β_{vq} indirectly controls the allocation of information by adjusting the token estimation accuracy. Specifically, a higher β_{vq} prioritize to minimize $\mathcal{L}_{vq}$, resulting in more information stored for token estimation. Conversely, a lower β_{vq} diminishes the weight of $\mathcal{L}_{vq}$, leading to allocating relatively more information for modifying VQGAN decoder features. As a result, β_{vq} adjusts the information allocation, controlling the trade-off between token estimation accuracy and image reconstruction fidelity.

In the next section, through our preliminary experiments, we will demonstrate the pivotal role of β_{vq} for optimal RDP performance. Furthermore, we will show that the ideal β_{vq} varies with bit rates, indicating the need to adjust β_{vq} for each target bit rate.

B. PRELIMINARY EXPERIMENTS

In this section, we conducted preliminary experiments to explore the impact of β_{vq} and β_{rate} on model performance.

We trained multiple models, each with fixed conditional inputs $(\beta_{vq}, \beta_{rate})$. Specifically, nine distinct models are trained using (11) with $(\beta_{vq}, \beta_{rate}) = (1.0, 1.76), (1.0, 0.73), (1.0, 0.16), (2.0, 1.9), (2.0, 1.12), (2.0, 0.63), (2.5, 2.02), (2.5, 1.47)$, and $(2.5, 0.84)$. The β_{rate} values were adjusted to ensure consistent average bit rates across the models. As a result, we obtained models for low (approximately 0.04 bits per pixel (bpp)), medium (0.1 bpp), and high (0.15 bpp) bit rates for $\beta_{vq} = \{1.0, 2.0, 2.5\}$, respectively. We used the OpenImage [63] dataset for training and the CLIC2020 [64] test dataset for evaluation. We evaluated the models using PSNR, FID [65], and LPIPS [62], which are widely used in generative LIC research. Given our focus on improving the RDP trade-off, a model performance that is well-balanced across these metrics is preferred. Other training details can be found in Sec. IV-A2.

Fig. 5 illustrates the results. We observed trends based on the value of β_{vq}. Specifically, a higher β_{vq} tends to yield better FID due to higher token estimation accuracy. In contrast, a lower β_{vq} results in improved PSNR and LPIPS, because more information is stored for decoder feature modification. Furthermore, the results indicate that different β_{vq} leads to the best RDP performance at different bit rates. At the lowest bit rate, the differences in PSNR and LPIPS among the three models are marginal; however, the model with $\beta_{vq} = 2.5$ achieves superior FID, resulting in the best overall performance. At the medium bit rate, the model with $\beta_{vq} = 2.0$ exhibits the most balanced performance across all three metrics. In contrast, at the highest bit rate, the model with $\beta_{vq} = 1.0$ achieves the best PSNR, FID, and LPIPS simultaneously, showcasing its superiority.

Based on these observations, we draw two main conclusions: (a) the selection of β_{vq} is pivotal for achieving a balanced RDP performance, and (b) it is essential to adjust β_{vq} for each specific target bit rate. However, exploring β_{vq} by training distinct models for each target bit rate leads to substantial computational costs. To address this challenge, we propose a dual-conditioned training strategy to adjust β_{vq} for each target bit rate without training multiple models.

C. DUAL-CONDITIONED TRAINING

In this section, we propose a novel **dual-conditioned training**. It is designed to train a single model that adjusts token reconstruction accuracy and bit rate based on the input β_{vq} and β_{rate}, respectively. It enables us to achieve optimal performance at each target bit rate by adjusting β_{rate} and β_{vq}, hence eliminating the need to train separate models using different $(\beta_{rate}, \beta_{vq})$ pairs.

As shown in Fig. 6, the training has three stages: (1) Dual conditioned pre-training stage, (2) β-Selection stage, and (3) Selected-β fine-tuning stage. During the first stage, the DC-VIC model is trained to adapt to a broad range of $(\beta_{vq}, \beta_{rate})$ values, enabling wide exploration in the following stage. The second stage involves the exploration and selection of optimal $(\beta_{rate}, \beta_{vq})$ for each target bit rate. Finally, in the third stage, we fine-tune the model with the selected $(\beta_{rate}, \beta_{vq})$ pairs from the second stage. We will explain the details below.

1) FIRST STAGE: DUAL CONDITIONED PRE-TRAINING

During this stage, we pre-train the DC-VIC model to minimize (11). The conditional inputs β_{rate} and β_{vq} are independently sampled at each training step from uniform distributions:

$$\beta_{rate} \sim U(0, \beta_{rate}^{\max})$$
$$\beta_{vq} \sim U(0, \beta_{vq}^{\max}), \tag{12}$$

where β_{rate}^{max} and β_{vq}^{max} are hyperparameters. As described in Sec. III-A, the conditional inputs are injected into the model via CFT layer and act as weights for the loss functions R and $\mathcal{L}_{vq}$ in (11). In this way, the model learns to adjust the bit rate and token reconstruction accuracy based on the given conditional inputs, β_{rate} and β_{vq}. Moreover, by sampling two inputs from independent uniform distributions at each iteration, the model adapts to various $(\beta_{vq}, \beta_{rate})$ pairs.

2) SECOND STAGE: β-SELECTION

Through the aforementioned Dual-conditioned pre-training, we can control the bit rate and token reconstruction accuracy with a single model by adjusting β_{rate} and β_{vq}. As discussed in Sec. III-B, selecting the appropriate β_{vq} for different bit rates is crucial for optimal performance. Thus, to attain optimal performance, we adjust β_{vq} for each target bitrate using a validation dataset. Specifically, given a set of target bit rates $\{r_1, r_2, \cdots, r_N\}$, we determine a pair $(\hat{\beta}_{rate}^{r_i}, \hat{\beta}_{vq}^{r_i})$ for each bit rate r_i as follows.

1) We define a set of β_{vq} candidates $\{\beta_{vq}^{(1)}, \beta_{vq}^{(2)}, \cdots, \beta_{vq}^{(M)}\}$. For each candidate $\beta_{vq}^{(j)}$, we perform a binary search to find a corresponding $\beta_{rate}^{(j)}$ such that the difference between the validation dataset's average bit rate and the target bit rate r_i is smaller than a threshold η.
2) Using the determined pairs $\{(\beta_{vq}^{(1)}, \beta_{rate}^{(1)}), \cdots, (\beta_{vq}^{(M)}, \beta_{rate}^{(M)})\}$, we generate M sets of reconstructions for the validation dataset.
3) We evaluate the distortion and perceptual quality of each reconstruction set using PSNR and FID. A weighted score S is then defined as:
$$S = \alpha \cdot \mathrm{PSNR} - \mathrm{FID}. \tag{13}$$
where α is a hyperparameter. For each of the M reconstruction sets, we calculate the score S, resulting in the set $\{S^{(1)}, S^{(2)}, \cdots, S^{(M)}\}$.
4) Among the M candidates, the pair $(\beta_{vq}^{(j)}, \beta_{rate}^{(j)})$ that achieves the highest score is selected as $(\hat{\beta}_{vq}^{r_i}, \hat{\beta}_{rate}^{r_i})$:
$$j = \underset{k \in \{1,2,\cdots,M\}}{\arg\max} \; S^{(k)}, \tag{14}$$
$$(\hat{\beta}_{vq}^{r_i}, \hat{\beta}_{rate}^{r_i}) = (\beta_{vq}^{(j)}, \beta_{rate}^{(j)}), \tag{15}$$

Repeating this procedure for each target bit rate r_i results in N optimal pairs of conditional inputs $\{(\hat{\beta}_{vq}^{r_1}, \hat{\beta}_{rate}^{r_1}), \cdots, (\hat{\beta}_{vq}^{r_N},$

TABLE 1. Bjøntegaard delta (BD) metrics [67] relative to ILLM [12]. Bold and underline indicate the best and second best results, respectively.

	BD-PSNR ↑	BD-FID ↓	BD-LPIPS ↓
ILLM [12]	0.0000	0.0000	0.00000
DIRAC [68]	-0.0188	1.2353	0.01638
CRDR [10]	**0.8845**	0.8164	0.00460
HiFiC [13]	-0.1511	1.0103	0.00190
HFD [11]	-0.4771	0.4828	-
Multi-Realism [9]	0.8390	0.9441	-
DC-VIC (PatchGAN)	-0.0996	-0.1761	-0.00141
DC-VIC (OASIS)	-0.0240	**-0.2754**	**-0.00303**

$\hat{\beta}^{r_N}_{rate})\}$. These selected pairs are used in the following fine-tuning stage and for evaluation. This β-selection allows us to find an ideal β_{vq} for each target bit rate within a single model, leading to substantial cost savings compared to training M distinct models using different β_{vq} values. Note that the weight of the model is fixed during this stage.

3) THIRD STAGE: SELECTED-β FINE-TUNING

In the previous stage, we determined optimal pairs $\{(\hat{\beta}^{r_1}_{vq}, \hat{\beta}^{r_1}_{rate}), \cdots, (\hat{\beta}^{r_N}_{vq}, \hat{\beta}^{r_N}_{rate})\}$ for N target bit rates. However, the model at this point is not fully optimized for these specific pairs because it is trained to handle a wide range of $(\beta_{vq}, \beta_{rate})$ in the first stage. To improve performance for the selected β pairs, we introduce a selected-β fine-tuning. During this stage, we fine-tune the model using the same loss function as in the pre-training stage, as defined in (11). However, instead of sampling conditional inputs from uniform distributions, we randomly sample a $(\hat{\beta}^{r_j}_{vq}, \hat{\beta}^{r_j}_{rate})$ pair from the selected N pairs at each training iteration. In this way, the model "forgets" its adaptability to various β pairs and enhances its performance on the N chosen conditional inputs.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUPS

1) DATASETS

We trained DC-VIC using the subset of the OpenImage [63] dataset. In the β-selection stage, we used 2000 images sampled from the OpenImage validation dataset. For the evaluation, we used three popular benchmark datasets: CLIC2020 [64], DIV2K [69], and Kodak [70].

2) IMPLEMENTATION DETAILS

For the pre-trained VQGAN [17], we used a checkpoint with a codebook size of $K = 256$, and a down-scale factor $f = H/h = 8$, available in the official implementation.[1] The results on different VQGAN configurations will be discussed in Sec. V-C.

During training, we employed random cropping and random flipping to obtain 256×256 patches from the original image. The batch size was set to six. For the loss function in (11), we set $\lambda_R = 0.5, \lambda_{img} = 1.0, \lambda^{img}_{MSE} = 50, \lambda_{vq} = 0.003, \lambda^{vq}_{MSE} = 2.0$, and $\lambda_{adv} = 0.01$.

In the dual-conditioned pre-training stage, The total number of training steps was set to $1,500,000$. In this stage, we followed the two-step training used in [12] and [71]. Specifically, we train the model without the adversarial loss $\mathcal{L}_{adv}$ in (11) for the first $1,000,000$ steps. Then, we fixed the parameters of the compression encoder and the entropy model and trained the other modules with $\mathcal{L}_{adv}$ for $500,000$ steps. The upper bounds of the conditional inputs in (12) were set to $\beta^{\max}_{vq} = 3.5$, $\beta^{\max}_{rate} = 3.0$.

For the β-selection stage, we searched $(\beta_{vq}, \beta_{rate})$ for $N = 5$ target bit rates: $\{0.05, 0.075, 0.1, 0.125, 0.15\}$ bpp (bits per pixel) using the OpenImage validation dataset. We set the β_{vq} candidates as $\{0.0, 0.25, 0.5, 0.75, 1.0, \cdots, 3.25, 3.5\}$, in total of 15 values. We set $\eta = 0.001$, $\alpha = 2.0$.

In the selected-β fine-tuning stage, we fine-tuned the model for $500,000$ steps. As in the latter part of the dual-conditioned pre-training stage, we fixed the parameters of the compression encoder and the entropy model.

In the preliminary experiments, we used the same training setting as the dual-conditioned pre-training stage, except for using a fixed $(\beta_{rate}, \beta_{vq})$ pair for each model.

For optimization, we used Adam [72] optimizer, where the initial learning rate was set to 1.0×10^{-4}. The learning rate was decreased to 1.0×10^{-5} for the last $300,000$ steps of the first $1,000,000$ steps in the pre-training stage (*i.e.*, the part where the model is trained without adversarial loss) and the last $200,000$ steps of the selected β fine-tuning stage. We used the same optimizer, learning rate, and scheduling to train the discriminator.

3) EVALUATION METRICS

For evaluation, we used the standard metrics in generative LIC research, PSNR, FID [65], and LPIPS [62]. To calculate FID, we followed the protocol in [13]. We do not calculate FID in Kodak dataset, because it has only 24 images. We also report Bjøntegaard delta (BD) metrics [67], including BD-PSNR, BD-FID, and BD-LPIPS. BD metrics represent the average difference in metrics (e.g., PSNR and FID) between two methods over a range of bit rates. We used the bjontegaard Python library,[2] which calculates BD metrics using a logarithmic scale of bit rates.

4) BASELINES

We compared our model with state-of-the-art generative LIC models, *Multi-Realism* [9], *HiFiC* [13], *ILLM* [12], *HFD* [11], *CRDR* [10], *DIRAC* [68], *PQ-MIM* [73], and VQGAN-based LIC method, *Mao+* [24]. Among the above methods, only CRDR and DIRAC are variable-rate methods, where the single model can compress an image into different bit rates. Our DC-VIC is also a variable-rate method because it can handle N target bit rates. For the other models, distinct models are required for different bit rates. Additionally, we compared

[1] https://github.com/CompVis/taming-transformers

[2] https://github.com/FAU-LMS/bjontegaard

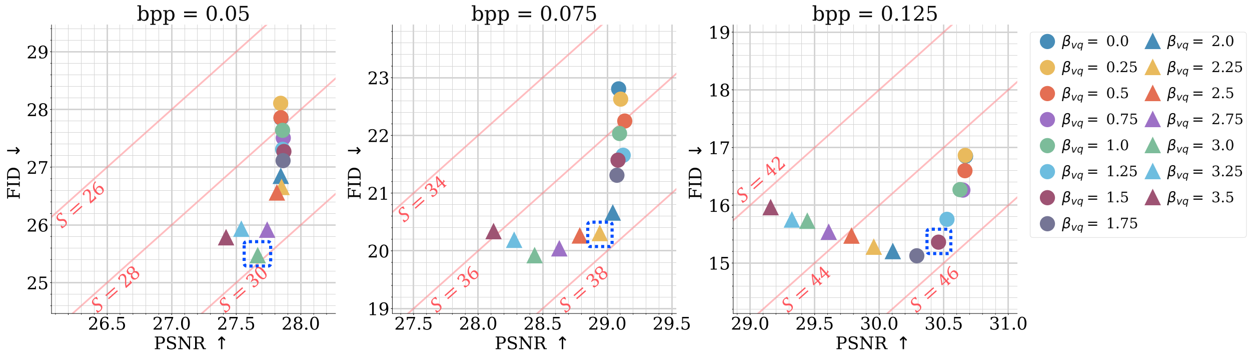


FIGURE 7. Results of the β-selection on three target bit rates, 0.05, 0.075, and 0.125 bpp on OpenImage validation dataset. Contour lines for score S are also shown. The marker highlighted with a blue dashed square is selected for the corresponding bit rate.

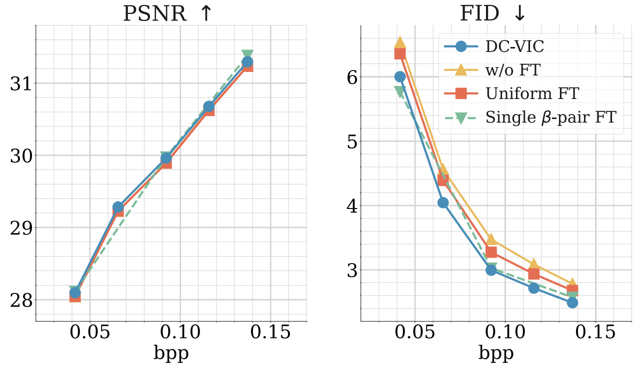


FIGURE 8. Comparison with various fine-tuning strategies on CLIC2020 test dataset. While there are no significant differences on PSNR, fine-tuning strategies affect FID.

our model with non-learned codecs, VVC [7] and BPG [66]. Note that these codecs target rate-distortion performance, whereas generative LIC methods, including DC-VIC, are designed to optimize rate-distortion-perception performance.

B. RESULTS OF β-SELECTION

In this section, we show the results of the β-Selection stage using the OpenImage validation dataset. Fig. 7 illustrates the PSNR-FID trade-off for $M = 15$ different β_{vq} candidates across three target bit rates, $\{0.05, 0.075, 0.125\}$ bpp. As detailed in Sec. III-C2, β_{rate} is adjusted for each β_{vq} so that the difference between the average bit rate and the target rate is less than η, allowing fair comparison among various β_{vq} values. The figure also highlights β_{vq} values selected using the score S in (13). The variation in PSNR and FID scores with β_{vq} indicates the significant influence of β_{vq} on compression performance. Generally, higher β_{vq} values tend to result in lower (better) FID, while lower β_{vq} values yield higher PSNR. This trend can be attributed to the information allocation between token estimation and decoder feature modification controlled by β_{vq}. Specifically, with a higher β_{vq}, more information is allocated to token estimation. It enables the VQGAN decoder to reconstruct the image from a highly accurate token sequence, thus enhancing perceptual quality (FID). Conversely, a lower β_{vq} prioritizes information for decoder feature modification, which enhances reconstruction fidelity, resulting in higher PSNR scores.

Next, we look into the results at specific target bit rates. At 0.05 bpp, a range of β_{vq} values yield similar PSNR scores; however, higher β_{vq} values lead to lower FID, thereby achieving a better PSNR-FID trade-off. As a result, $\beta_{vq} = 3.0$ is selected based on the weighted score S. On the other hand, at 0.125 bpp, higher β_{vq} values tend to result in worse FID scores. Thus, lower β_{vq} values achieve a more favorable PSNR-FID trade-off, leading to the selection of $\beta_{vq} = 1.5$ for this bit rate. These results indicate that higher β_{vq} values are preferable at lower bit rates, while lower β_{vq} values are more beneficial at higher bit rates. Additionally, this demonstrates the effectiveness of adopting different β_{vq} values for each target bit rate, thereby validating the core concept of our dual-conditioned training.

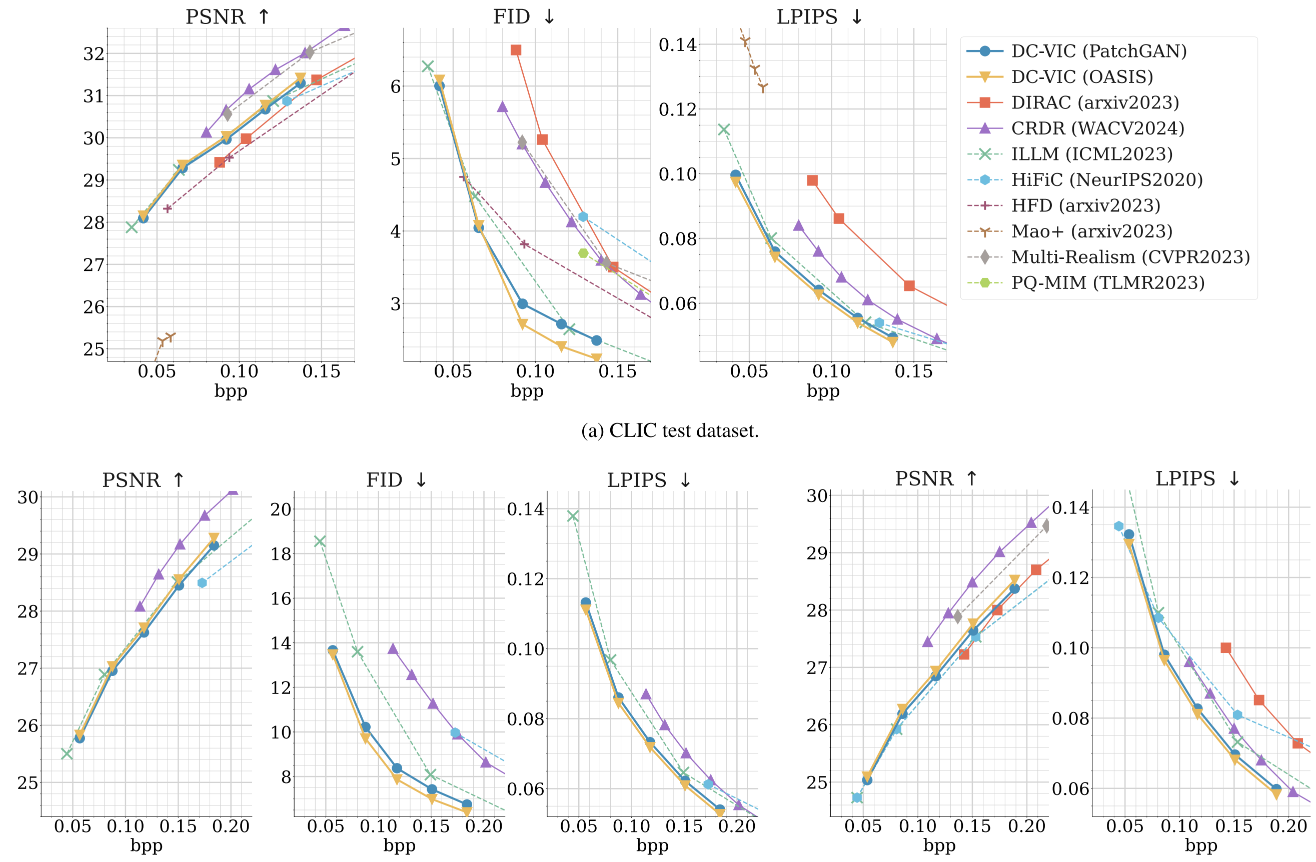
C. EFFECTIVENESS OF SELECTED β FINE-TUNING

In DC-VIC, the selected-β fine-tuning is adopted to optimize the model for the selected pairs $\{(\tilde{\beta}_{vq}^{r_1}, \tilde{\beta}_{rate}^{r_1}) \cdots (\tilde{\beta}_{vq}^{r_N}, \tilde{\beta}_{rate}^{r_N})\}$. In this section, we evaluate the efficacy of our fine-tuning approach by comparing it with three baseline strategies:

- *w/o fine-tuning (FT)*: This baseline uses the model as it is after the dual-conditioned pre-training stage, without any fine-tuning. During the evaluation, we use $(\tilde{\beta}_{vq}^{r_i}, \tilde{\beta}_{rate}^{r_i})$ pairs determined through β-selection.
- *Uniform FT*: This approach is equal to extending the first stage of training. Specifically, the model is fine-tuned with $(\beta_{vq}, \beta_{rate})$ pairs sampled from a uniform distribution. Subsequent to this fine-tuning, we select the optimal $(\tilde{\beta}_{vq}^{r_i}, \tilde{\beta}_{rate}^{r_i})$ pairs through β-selection and evaluate the model's performance using these pairs.
- *Single β-pair FT*: This strategy involves fine-tuning the model using only one of the selected $(\tilde{\beta}_{vq}^{r_i}, \tilde{\beta}_{rate}^{r_i})$ pairs, thereby optimizing the model for a single target bit rate.

To save computational cost, we trained three distinct models for the *Single β-pair FT*, using the target bit rates $\{0.05, 0.10, 0.15\}$ bpp.

The results on the CLIC2020 test dataset are illustrated in Fig. 8. As shown in the figure, fine-tuning



(a) CLIC test dataset.

(b) DIV2K dataset.

(c) Kodak dataset.

FIGURE 9. Quantitative comparison with state-of-the-art generative LIC methods on (a) CLIC, (b) DIV2K, and (c) Kodak datasets. The solid lines indicate variable-rate methods, while the dashed lines represent single-rate methods, which require separate models for each bit rate. Our DC-VIC achieves the best FID and LPIPS scores across all three datasets. FID is not calculated on the Kodak dataset due to its limited number of images.

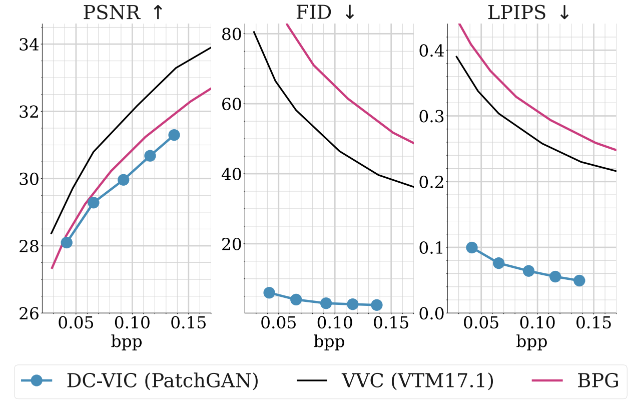


FIGURE 10. Quantitative comparison with non-learned codecs (VVC [7] and BPG [66]) on the CLIC2020 test dataset. While these codecs outperform DC-VIC in terms of PSNR, they perform significantly worse on perceptual metrics, FID and LPIPS .

methods have a notable impact on FID scores, while PSNR scores are unaffected. The comparison between *Uniform-sampling FT* and *w/o FT* shows only a marginal improvement in FID, suggesting that merely extending the training duration does not significantly enhance performance. In contrast, DC-VIC achieves superior FID scores over *Uniform-sampling FT* across all target bit rates. This indicates that the generalized training with uniformly sampled $(\beta_{vq}, \beta_{rate})$ pairs in *Uniform-sampling FT* led to suboptimal performance, which demonstrates the effectiveness of our fine-tuning using only the selected conditional inputs. Furthermore, the FID scores of DC-VIC are comparable to those of *Single β-pair FT*. The *Single β-pair FT* is a special case of DC-VIC, where only one target rate is specified at the β-selection stage. Hence, these small performance gaps indicate DC-VIC's robustness across various bit rates without sacrificing quality. Overall, the results validate the advantage of our selected-β fine-tuning in optimally adjusting the model for the selected conditional inputs, $\{(\tilde{\beta}_{vq}^{r_1}, \tilde{\beta}_{rate}^{r_1}) \cdots (\tilde{\beta}_{vq}^{r_N}, \tilde{\beta}_{rate}^{r_N})\}$.

D. QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS

We benchmarked our DC-VIC against the state-of-the-art (SOTA) generative LIC methods on three datasets, CLIC [64], DIV2K [69], and Kodak [70], in Fig. 9. Note that not all methods have reported results on the DIV2K and Kodak datasets. We present two variants of our model: DC-VIC (PatchGAN) and DC-VIC (OASIS), using PatchGAN and OASIS discriminators, respectively. In the figure, the dashed

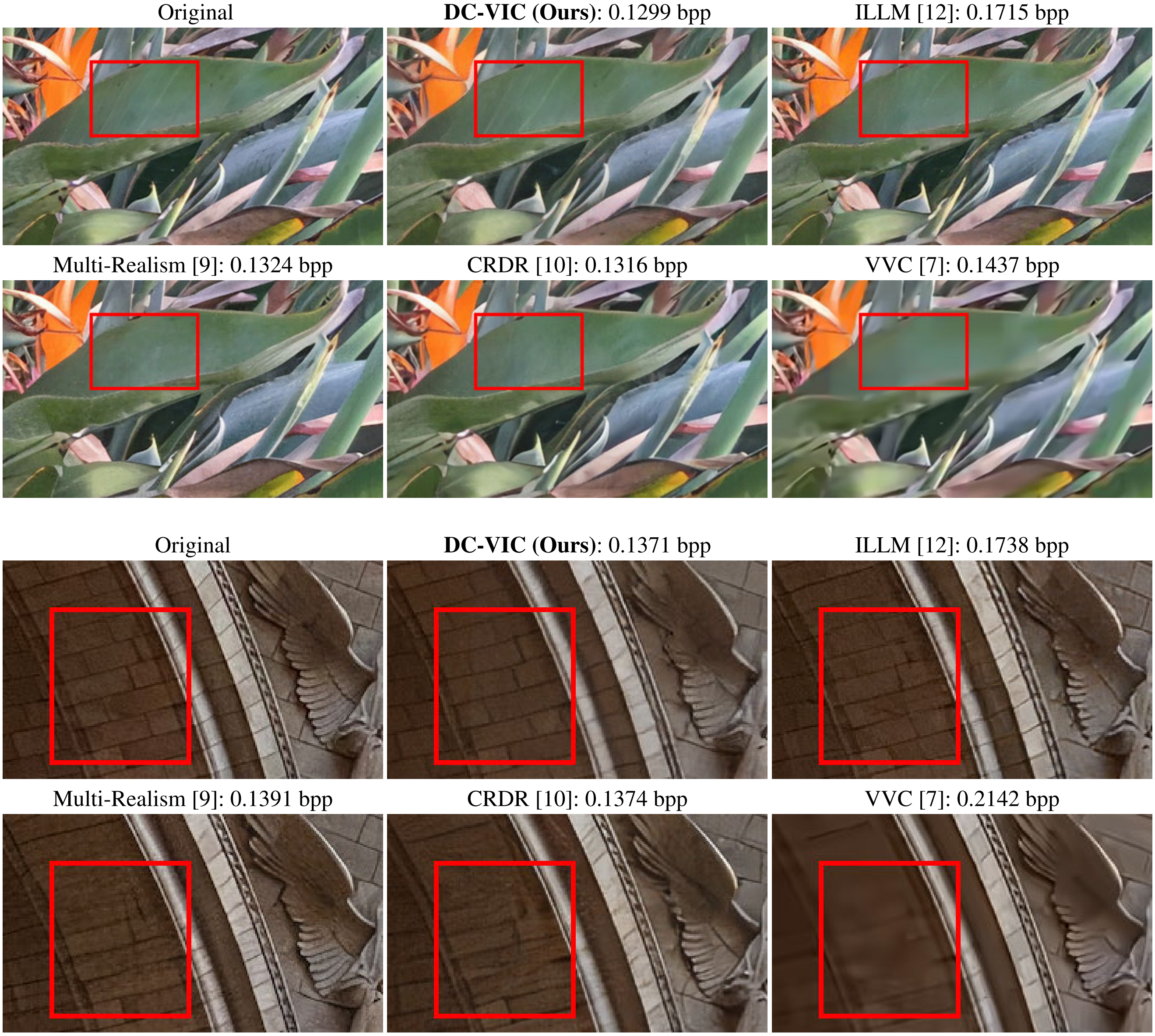


FIGURE 11. Qualitative comparison between our DC-VIC and state-of-the-art generative LIC methods on CLIC2020 dataset [64].

lines represent single-rate methods, and the solid lines in the figure denote variable-rate methods, including DC-VIC. While single-rate methods require separate models for each bit rate, variable-rate methods can handle different bit rates within a single model.

The results demonstrate that DC-VIC (PatchGAN) outperforms SOTA methods in perceptual metrics, FID and LPIPS, across all datasets, while achieving comparable PSNR to ILLM [12]. Notably, DC-VIC achieves this performance using a single model across different bit rates, whereas most other methods require separate models for each bit rate. Although CRDR [10] and Multi-Realism [9] surpass DC-VIC in terms of PSNR, our model achieves significantly better FID scores. Moreover, DC-VIC (OASIS) further enhances LPIPS and FID at specific bit rates, indicating the effectiveness of incorporating the OASIS discriminator into our method. In addition, DC-VIC significantly outperforms another pre-trained VQGAN-based method, Mao+ [24], in both PSNR and LPIPS, with a notable difference of over 2.5 dB in PSNR. This substantial margin suggests that Mao+ suffers from poor reconstruction fidelity, likely due to directly using VQGAN reconstructions as the final output. In contrast, our method mitigates artifacts and distortions introduced by VQGAN through feature modification, leading to improved reconstruction fidelity. Note that we can not compare FID due to different FID calculation protocols and the lack of publicly available implementation.

We also report the Bjøntegaard delta metrics [67] (BD-PSNR, BD-FID, and BD-LPIPS) for generative LIC methods relative to ILLM [12] in Table 1. As shown in the table, DC-VIC (PatchGAN) and DC-VIC (OASIS) achieved the best and second-best BD-FID and BD-LPIPS, respectively,

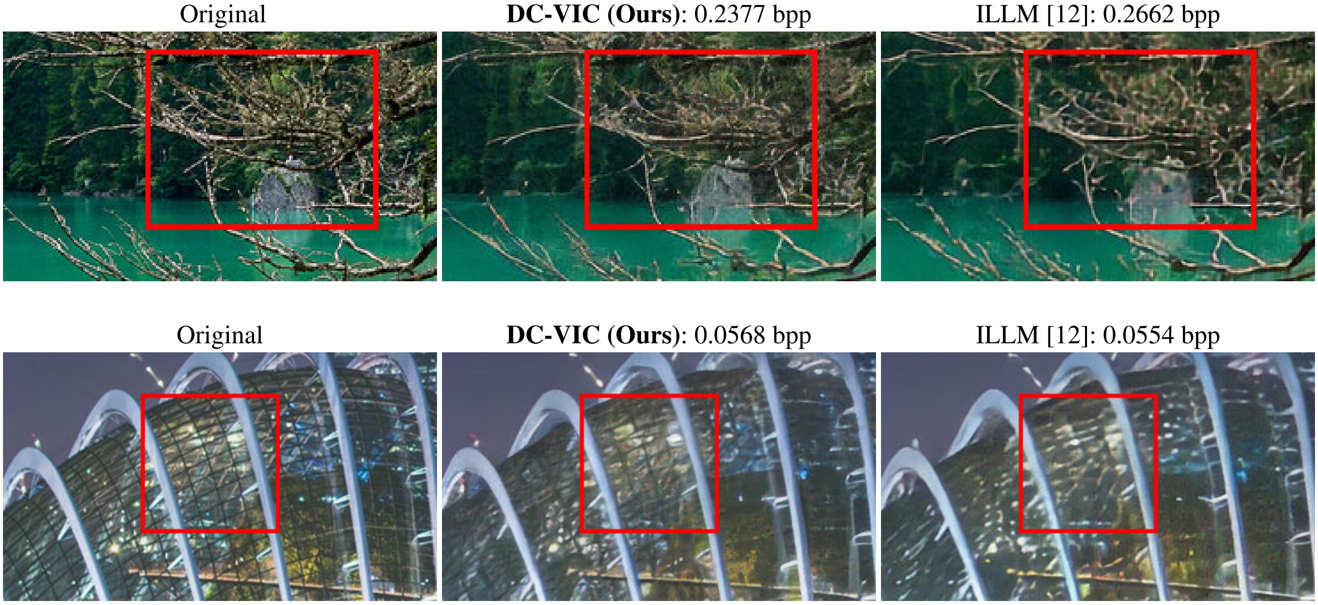


FIGURE 12. Qualitative comparison between DC-VIC and ILLM [12] on DIV2K dataset [69].

demonstrating superior performance in perceptual metrics. While the negative BD-PSNR values indicate that both DC-VIC models have slightly lower PSNR than ILLM on average, the differences are minimal: 0.0996 dB for DC-VIC (PatchGAN) and just 0.024 dB for DC-VIC (OASIS). This indicates competitive PSNR relative to ILLM.

Furthermore, we compare DC-VIC and non-learned codecs, VVC [7] and BPG [66] in Fig. 10. Since VVC and BPG focus on rate-distortion performance, they outperform DC-VIC in PSNR, a distortion metric. However, they perform significantly worse in terms of LPIPS and FID, indicating poorer perceptual quality.

In conclusion, both DC-VIC (PatchGAN) and DC-VIC (OASIS) consistently achieve superior rate-distortion-perception (RDP) performance across all three datasets compared to state-of-the-art generative LIC methods, including both single- and variable-rate models. DC-VIC also significantly outperforms traditional non-learned codecs in perceptual metrics.

E. QUALITATIVE COMPARISON

TABLE 2. Lossless bit rates of the three different VQGAN configurations that we tested.

Config.	Down-scale factor f	Codebook size N	Lossless bit rate [bpp]
(a)	8	256	0.125
(b)	8	16384	0.219
(c)	16	16384	0.0547

Fig. 11 shows a qualitative comparison between our DC-VIC and other SOTA methods, ILLM [12], Multi-realism [9], CRDR [10], and VVC [7] on CLIC2020 test dataset [64]. As shown in the figure, the reconstructions of DC-VIC exhibit fewer artifacts than other methods and retain the original image content with higher fidelity. For example, in the top image, DC-VIC successfully captures the leaf veins, whereas other methods failed to reconstruct them. Similarly, in the bottom examples, while other methods introduce artifacts, as marked with the red rectangles, DC-VIC accurately preserves the brick patterns without noticeable distortion. Notably, the decoded images from VVC [7] appear blurry, even at a higher bit rate than other methods. This results in poor perceptual quality, as reflected in the inferior perceptual metrics shown in Fig. 10.

In Fig. 12, we further compare the reconstructions of DC-VIC and ILLM [12] on DIV2K dataset [69]. In the top image, ILLM generates artifacts, as highlighted by the red rectangle. In contrast, the texture of our reconstruction appears more natural. In the bottom example, DC-VIC reconstructs finer structural details of the building, resulting in more accurate reconstruction than ILLM.

We hypothesize that utilizing the pre-trained VQGAN codebook contributed to the precise preservation of patterned textures, while the feature modification resulted in reduced artifacts. The presented results indicate DC-VIC's superior reconstruction quality across various datasets, demonstrating its effectiveness in reconstructing detailed image content and minimizing artifacts.

V. DISCUSSION

A. COMPARISON WITH ORIGINAL VQGAN RECONSTRUCTION

In this analysis, we compare the reconstructions of DC-VIC with those from the original VQGAN [17]. Specifically, we use the pre-trained VQGAN decoder to reconstruct images directly from the original VQGAN tokens z_q without any decoder feature modification. The original image and their reconstructions are shown in Fig. 13. As can be seen, the original VQGAN reconstructions exhibit significant

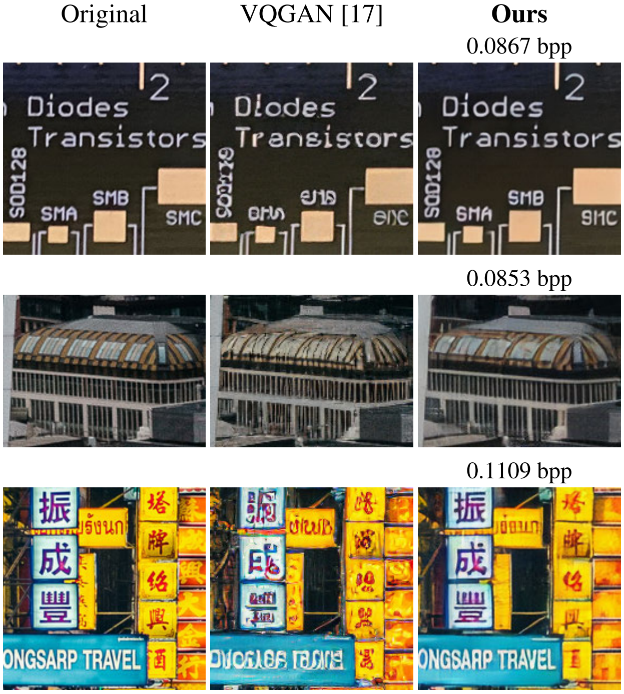


FIGURE 13. Comparison between standard VQGAN reconstructions (i.e., decoded from original VQGAN tokens without decoder feature modification) and our reconstructions. Noticeable distortions and artifacts are present in the VQGAN reconstructions, whereas these artifacts are mitigated in our reconstructions.

distortion and artifacts, despite using the original tokens. This indicates that although the VQGAN codebook is capable of capturing a wide range of image content, its reconstruction fidelity remains constrained. In contrast, our method modifies the decoder's intermediate features, enabling more accurate reconstructions. This improvement suggests that our decoder feature modification substantially enhances reconstruction fidelity, addressing the inherent limitation of the original VQGAN.

B. ANALYSIS ON TOKEN RECONSTRUCTION ACCURACY

As demonstrated in Sec III-B, the best performance was achieved with different β_{vq} values at various target bit rates. It suggests that the ideal information allocations for VQGAN token estimation vary with bit rates. In this section, we investigate the accuracy of token estimation across different β_{vq} values to understand the relationship between token estimation accuracy and overall compression performance.

Fig. 14 presents PSNR and token estimation accuracy on the OpenImage validation dataset. Here, the token estimation accuracy (Acc) is defined as follows:

$$s_i = \begin{cases} 1, & \text{if } \tilde{I}_{vq}^{(i)} = I_{vq}^{(i)} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$\mathrm{Acc} = \frac{1}{h \times w} \sum_{i}^{h \times w} s_i, \tag{17}$$

where h, w denote the height and width of the VQGAN latent $\hat{z}$, respectively. As in the β-selection stage, we used $M = 15$ distinct β_{vq} values across $N = 5$ target bit rates. The results demonstrate that a higher β_{vq} leads to lower PSNR and higher Acc, whereas a lower β_{vq} results in higher PSNR and lower Acc. This observation aligns with the objective of our dual-conditioned training, demonstrating that β_{vq} effectively controls the information allocation between token estimation and decoder feature modification within a single model.

Moreover, red circles in Fig. 14 denote the results obtained using the selected conditional inputs: $\{(\tilde{\beta}_{vq}^{r_1}, \tilde{\beta}_{rate}^{r_1}), \cdots (\tilde{\beta}_{vq}^{r_N}, \tilde{\beta}_{rate}^{r_N})\}$. Interestingly, the token estimation accuracy of these selected conditional inputs converged around 0.3. This suggests that, even at relatively higher bit rates, high token estimation accuracy is not necessary for optimal RDP performance. We hypothesize that token estimation accuracy exceeding a certain threshold does not significantly enhance the reconstruction quality of the VQGAN. As a result, conditional inputs that allocated more information for decoder feature modification were favored at higher bit rates.

Furthermore, this insight could be useful in designing a more efficient training strategy. For instance, if higher token estimation accuracy is unnecessary, we might consider limiting the range of conditional inputs in the initial training stage. This adjustment could enable the model to focus on a narrower set of inputs, potentially enhancing performance.

C. TRAINING WITH DIFFERENT VQGAN CONFIGURATIONS

In our main experiments, we used a pre-trained VQGAN model with a down-scale factor $f = 8$ and a codebook size of $K = 256$. In this section, we evaluate our DC-VIC trained with other pre-trained VQGAN models, specifically $(f, K) = (8, 16384)$ and $(16, 16384)$, to evaluate the robustness. These pre-trained models are also available in the author's repository.[3],[4]

First, to understand the characteristics of these configurations, we present the lossless bit rates of the token sequence in Table 2. The lossless bit rate can be calculated as $\log_2 K/f^2$, where $\log_2 K$ denotes a bit length required to represent each token, and f^2 corresponds to the number of pixels covered by a single token. It shows that $(f, K) = (8, 16384)$ and $(f, K) = (16, 16384)$ have the highest and lowest lossless compression bit rates, respectively.

We report the PSNR and FID scores of DC-VIC across different VQGAN configurations in Fig. 15. The figure indicates that different configurations excel at different bit rates. For example, while the $(f, K) = (16, 16384)$ model outperforms the $(f, K) = (8, 256)$ model at the lowest bit rate, it resulted in poor FID at higher bit rates. Conversely, while the $(f, K) = (8, 16384)$ configuration shows less optimal FID at lower bit rates, it demonstrates competitive performance at higher bit rates compared to our default configuration. We hypothesize that these trends are related to

[3] https://ommer-lab.com/files/latent-diffusion/vq-f8.zip

[4] https://heibox.uni-heidelberg.de/f/0e42b04e2e904890a9b6/?dl=1

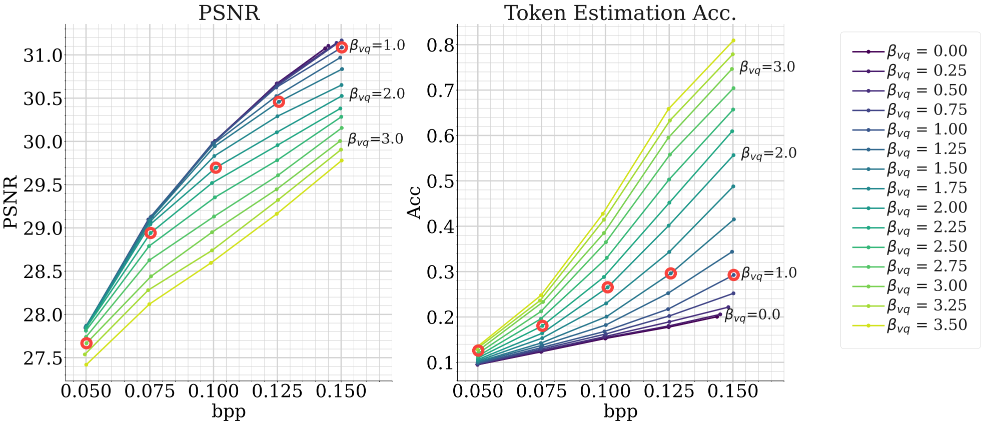


FIGURE 14. PSNR and token estimation accuracy on OpenImage Validation dataset [63] using the model before the Selected-β fine-tuning stage. We used $M = 15$ β_{vq} values for $N = 5$ target bit rates, obtaining 75 data points in total. Red circles indicate the results from selected β pairs: $(\tilde{\beta}_{vq}^{r_i}, \tilde{\beta}_{rate}^{r_i})$.

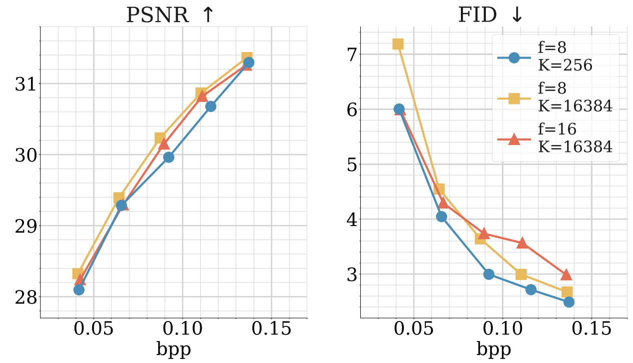


FIGURE 15. Quantitative results on three different VQGAN configurations on CLIC2020 test dataset. $(f, K) = (8, 256)$ is our default setting.

the lossless compression rate. The $(f, K) = (16, 16384)$ and $(8, 16384)$ models represent the lowest and highest lossless compression rates among the three configurations, possibly leading to their superior performance at low and high bit rates, respectively. Furthermore, our default setting $(f, K) = (8, 256)$ consistently achieves favorable performance. One possible reason for this is its relatively smaller codebook size, which simplifies our token estimator's task of predicting indices for K classes. The larger $K = 16384$ in the other configurations may complicate the training of the token estimator, potentially leading to challenges in optimization.

These findings also imply that while DC-VIC achieves robust performance across various VQGAN variations, there may exist even more optimal VQGAN configurations for specific target bit rates. While the pre-training of VQGAN extends beyond the scope of this study, investigating more customized VQGAN configurations could further enhance RDP performance, presenting a promising direction for our future research.

Furthermore, our model can be used alongside any other codebook-based generative models if SFT layers can be integrated into the decoder, which is feasible for most models. Since the original VQGAN [17], several improved versions have been proposed [74], [75], [76], [77]. While these methods introduce enhancements, such as Transformer-based autoencoder architectures [75], [76] or improved codebooks [74], [77], their fundamental concept, reconstructing high-quality images from discrete token sequences, remains consistent with the original VQGAN [17]. Therefore, as our method demonstrates robust performance across various VQGAN configurations, it is expected to perform effectively with other codebook-based generative models as well.

TABLE 3. Comparison of parameter counts and encoding/decoding times between our DC-VIC and existing methods. The encoding and decoding times are measured using images with a spatial resolution of 256 × 256. The encoding and decoding times for DIRAC [68] are omitted due to the lack of publicly available implementation.

Method	Parameter count	Enc. time [ms]	Dec. time [ms]
CRDR [10]	127.8M	38.30	67.64
DIRAC [68]	136.8M	-	-
ILLM [12]	181.5M	35.03	32.87
DC-VIC	151.7M (84.1M excluding VQGAN [17])	39.66	86.86

D. MODEL SIZE AND COMPUTATIONAL COMPLEXITY

Table 3 presents a comparison of model sizes and encoding/decoding times. Note that some methods do not report their model sizes or publish their implementations. Therefore, their results are omitted. To measure the average encoding and decoding times, we used 100 images from the DIV2K [69] dataset, cropping them to obtain 256 × 256 patches. The experiments were conducted on a machine equipped with an NVIDIA GeForce RTX 3080 GPU (CUDA version 11.6), an AMD Ryzen 7 3700X 8-Core Processor, running Ubuntu 18.04.6 LTS.

In terms of parameter counts, we observe that there are no significant differences between DC-VIC and other methods. This indicates that the performance improvements of our method are not merely due to an increase in parameters. Additionally, the pre-trained VQGAN [17], with 67.6M parameters, remains fixed in our model. Consequently, the learnable part of our method consists of 84.1M parameters, which is fewer than the number of parameters in other methods.

Regarding encoding speed, there are no notable differences between CRDR [10], ILLM [12], and DC-VIC. However, DC-VIC exhibits the slowest decoding speed among the three methods. Concretely, our model is 19.22 ms slower than CRDR and 53.99 ms slower than ILLM. This slower performance is attributed to the relatively high computational cost of the VQGAN decoder. A potential solution to this issue is incorporating a more efficient variant of VQGAN, such as Efficient VQGAN [76].

In terms of total training time, our approach is expected to be shorter than existing single-rate methods. Since our model is variable-rate, we do not need to train multiple models for different bit rates. Our single model is optimized for five different bit rates, whereas single-rate methods such as ILLM [12] and HiFiC [13] require training five distinct models, leading to longer overall training times. Additionally, our total training process has 2 million iterations, which is comparable to or shorter than other methods (e.g., HiFiC [13] and Multi-Realism [9] uses 2 million and 3 million iterations per model, respectively).

VI. CONCLUSION

In this paper, we introduced a Dual-Conditioned VQGAN-based Image Compression (DC-VIC) model. On the encoder side, the original image and its VQGAN tokens are integrated into a single bitstream. On the decoder side, the image is reconstructed through VQGAN token estimation and VQGAN decoder feature modification. Preliminary experiments demonstrated that the information allocation between these two processes significantly influences compression performance, and the optimal allocations vary with target bit rates. To dynamically control the information allocation within a single model, we proposed a Dual-Conditioned Training approach. We initially train the model to adapt to a wide range of conditional inputs, which adjust the total bit rate and information allocation. Subsequently, we employ a β-Selection protocol to identify the optimal conditional inputs for each target bit rate. The model is then fine-tuned with these selected inputs.

We empirically demonstrated the effectiveness of the β-Selection and fine-tuning strategy. In addition, DC-VIC outperformed existing state-of-the-art generative LIC methods in terms of perceptual metrics. Furthermore, we investigated the token estimation accuracy of the selected conditional inputs and evaluated the model's robustness across different pre-trained VQGAN configurations. These explorations provide insights for enhancing model performance and suggest promising directions for future research.

LIMITATIONS AND FUTURE WORK

Currently, there are two main limitations in our work. Firstly, our method is focused on low bit rate compression (below 0.20 bpp). However, for practical application, achieving robust performance across a wide range of bit rates is essential. One potential solution to address this limitation is to employ a pre-trained VQGAN with a larger codebook and a smaller down-scaling factor. Although this would increase the data size of the token sequence, it is expected to enhance reconstruction quality, making the method more suitable for higher bit rate compression.

Secondly, as discussed in Sec. V-D, the decoding process of DC-VIC is slower compared to other GAN-based methods [10], [12]. Since image compression is a fundamental process, minimizing computational cost is essential. To mitigate this issue, we could incorporate improved versions of VQGAN, such as MoVQ [74] and Efficient VQGAN [76]. Since our method is compatible with any VQGAN architecture, it can easily incorporate these variants. For example, Efficient VQGAN [76] replaces the global attention modules with the Swin Transformer Block [36], achieving faster reconstruction and enhanced quality. Adopting such enhanced VQGAN models is a promising direction for future work to improve both the efficiency and compression performance of our method.

Additionally, developing distortion-perception controllability, as in [9] and [10], is another interesting direction for future work. As shown in Fig. 7, β_{vq} adjusts trade-off between PSNR (distortion) and FID (perception) to some extent. Although the current controllable range is narrower than that of existing methods [9], [10], this phenomenon suggests that our model has the potential to develop distortion-perception controllability. Such a feature would be valuable in real-world scenarios, as it would allow users to select the preferred level of realism based on specific requirements.

REFERENCES

[1] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[2] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[3] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3339–3343.

[4] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5708–5717.

[5] J. Liu, H. Sun, and J. Katto, "Learned image compression with mixed transformer-CNN architectures," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14388–14397.

[6] F. Mentzer, E. Agustson, and M. Tschannen, "M2T: Masking transformers twice for faster decoding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 5317–5326.

[7] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[8] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 221–231.

[9] E. Agustsson, D. Minnen, G. Toderici, and F. Mentzer, "Multi-realism image compression with a conditional generator," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22324–22333.

[10] S. Iwai, T. Miyazaki, and S. Omachi, "Controlling rate, distortion, and realism: Towards a single comprehensive neural image compression model," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 2888–2897.

[11] E. Hoogeboom, E. Agustsson, F. Mentzer, L. Versari, G. Toderici, and L. Theis, "High-fidelity image compression with score-based generative models," 2023, *arXiv:2305.18231*.

[12] M. J. Muckley, A. El-Nouby, K. Ullrich, H. Jégou, and J. Verbeek, "Improving statistical fidelity for neural image compression with implicit local likelihood models," in *Proc. 40th Int. Conf. Mach. Learn. (ICML)*, Jan. 2023, pp. 25426–25443.

[13] F. Mentzer, G. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 11913–11924.

[14] Y. Blau and T. Michaeli, "Rethinking lossy compression: The rate-distortion-perception tradeoff," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 675–685.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 6840–6851.

[17] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12868–12878.

[18] M. Xu, S. Li, J. Lu, and W. Zhu, "Compressibility constrained sparse representation with learnt dictionary for low bit-rate image compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1743–1757, Oct. 2014.

[19] X. Zhan, R. Zhang, D. Yin, and C. Huo, "SAR image compression using multiscale dictionary learning and sparse representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 5, pp. 1090–1094, Sep. 2013.

[20] J. Zepeda, C. Guillemot, and E. Kijak, "Image compression using sparse representations and the iteration-tuned and aligned dictionary," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 1061–1073, Sep. 2011.

[21] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "MaskGIT: Masked generative image transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11305–11315.

[22] J. Lezama, H. Chang, L. Jiang, and I. Essa, "Improved masked image generation with token-critic," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2022, pp. 70–86.

[23] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10686–10696.

[24] Q. Mao, T. Yang, Y. Zhang, Z. Wang, M. Wang, S. Wang, and S. Ma, "Extreme image compression using fine-tuned VQGANs," 2023, *arXiv:2307.08265*.

[25] W. Jiang, W. Wang, and Y. Chen, "Neural image compression using masked sparse visual representation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 4177–4185.

[26] D. Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 10794–10803.

[27] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[28] C. Li, S. Yin, C. Jia, F. Meng, Y. Tian, and Y. Liang, "Multirate progressive entropy model for learned image compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 8, pp. 7725–7741, Aug. 2024.

[29] Y. Qian, X. Sun, M. Lin, Z. Tan, and R. Jin, "Entroformer: A transformer-based entropy model for learned image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[30] W. Jiang, J. Yang, Y. Zhai, F. Gao, and R. Wang, "MLIC++: Linear complexity multi-reference entropy modeling for learned image compression," 2023, *arXiv:2307.15421*.

[31] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14766–14775.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[33] H. Fu, F. Liang, J. Liang, B. Li, G. Zhang, and J. Han, "Asymmetric learned image compression with multi-scale residual block, importance scaling, and post-quantization filtering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 4309–4321, Aug. 2023.

[34] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7936–7945.

[35] Z. Tang, H. Wang, X. Yi, Y. Zhang, S. Kwong, and C.-C. J. Kuo, "Joint graph attention and asymmetric convolutional neural network for deep image compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 421–433, Jan. 2023.

[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[37] R. Zou, C. Song, and Z. Zhang, "The devil is in the details: Window-based attention for image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17471–17480.

[38] T. Ladune, P. Philippe, F. Henry, G. Clare, and T. Leguay, "COOL-CHIC: Coordinate-based low complexity hierarchical image codec," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 13469–13476.

[39] H. Kim, M. Bauer, L. Theis, J. R. Schwarz, and E. Dupont, "C3: High-performance and low-complexity neural compression from a single image or video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 9347–9358.

[40] E. Dupont, H. Loya, M. Alizadeh, A. Goliński, Y. W. Teh, and A. Doucet, "COIN++: Neural compression across modalities," *Trans. Mach. Learn. Res.*, Jan. 2022. [Online]. Available: https://jmlr.org/tmlr/papers/

[41] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6228–6237.

[42] M. Careil, M. J. Muckley, J. Verbeek, and S. Lathuilière, "Towards image compression with perfect realism at ultra-low bitrates," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[43] T. Xu, Z. Zhu, D. He, Y. Li, L. Guo, Y. Wang, Z. Wang, H. Qin, Y. Wang, J. Liu, and Y.-Q. Zhang, "Idempotence and perceptual image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[44] S. Iwai, T. Miyazaki, and S. Omachi, "Semantically-guided image compression for enhanced perceptual quality at extremely low bitrates," *IEEE Access*, vol. 12, pp. 100057–100072, 2024.

[45] Z. Jia, J. Li, B. Li, H. Li, and Y. Lu, "Generative latent coding for ultra-low bitrate image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 26088–26098.

[46] N. Körber, E. Kromer, A. Siebert, S. Hauke, D. Mueller-Gritschneder, and B. Schuller, "EGIC: Enhanced low-bit-rate generative image compression guided by semantic segmentation," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2025, pp. 202–220.

[47] A. van den Oord, O. Vinyals, and k. Kavukcuoglu, "Neural discrete representation learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 6309–6318.

[48] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 606–615.

[49] S. Zhou, K. C. Chan, C. Li, and C. C. Loy, "Towards robust blind face restoration with codebook lookup transformer," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 30599–30611.

[50] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2020, pp. 405–421.

[51] H. Li, S. Li, S. Ding, W. Dai, M. Cao, C. Li, J. Zou, and H. Xiong, "Image compression for machine and human vision with spatial-frequency adaptation," 2024, *arXiv:2407.09853*.

[52] J. Liu, R. Feng, Y. Qi, Q. Chen, Z. Chen, W. Zeng, and X. Jin, "Rate-distortion-cognition controllable versatile neural image compression," 2024, *arXiv:2407.11700*.

[53] Y.-H. Chen, Y.-C. Weng, C.-H. Kao, C. Chien, W.-C. Chiu, and W.-H. Peng, "TransTIC: Transferring transformer-based image compression from human perception to machine perception," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 23240–23250.

[54] Z. Cui, J. Wang, S. Gao, T. Guo, Y. Feng, and B. Bai, ''Asymmetric gained deep image compression with continuous rate adaptation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10527–10536.

[55] Y. Choi, M. El-Khamy, and J. Lee, ''Variable rate deep image compression with a conditional autoencoder,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3146–3154.

[56] S. Cai, Z. Zhang, L. Chen, L. Yan, S. Zhong, and X. Zou, ''High-fidelity variable-rate image compression via invertible activation transformation,'' in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2021–2031.

[57] M. Song, J. Choi, and B. Han, ''Variable-rate deep image compression through spatially-adaptive feature transform,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2360–2369.

[58] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, ''High-resolution image synthesis with latent diffusion models,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[59] C. Chen, X. Shi, Y. Qin, X. Li, X. Han, T. Yang, and S. Guo, ''Real-world blind super-resolution via feature matching with implicit high-resolution priors,'' in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 1329–1338.

[60] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, ''Image-to-image translation with conditional adversarial networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[61] E. Schönfeld, V. Sushko, D. Zhang, J. Gall, B. Schiele, and A. Khoreva, ''You only need adversarial supervision for semantic image synthesis,'' in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[62] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, ''The unreasonable effectiveness of deep features as a perceptual metric,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[63] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, ''The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale,'' in *Proc. Int. J. Comput. Vis.*, vol. 128, Jul. 2020, pp. 1956–1981.

[64] G. Toderici, L. Theis, N. Johnston, E. Agustsson, F. Mentzer, J. Balle, W. Shi, and R. Timofte. (2020). *CLIC 2020: Challenge on Learned Image Compression*. [Online]. Available: https://www.tensorflow.org/datasets/catalog/clic

[65] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, ''GANs trained by a two time-scale update rule converge to a local Nash equilibrium,'' in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 6629–6640.

[66] F. Bellard. (2014). *BPG Image Format*. [Online]. Available: https://bellard.org/bpg/

[67] G. Bjøntegaard. (2001). *Calculation of Average PSNR Differences Between RD-Curves*. [Online]. Available: https://api.semanticscholar.org/CorpusID

[68] N. Fathima Ghouse, J. Petersen, A. Wiggers, T. Xu, and G. Sautière, ''A residual diffusion model for high perceptual quality codec augmentation,'' 2023, *arXiv:2301.05489*.

[69] E. Agustsson and R. Timofte, ''NTIRE 2017 challenge on single image super-resolution: Dataset and study,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131.

[70] (1991). *Kodak Photodc Dataset*. [Online]. Available: https://r0k.us/graphics/kodak

[71] S. Iwai, T. Miyazaki, Y. Sugaya, and S. Omachi, ''Fidelity-controllable extreme image compression with generative adversarial networks,'' in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 8235–8242.

[72] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[73] A. El-Nouby, M. J. Muckley, K. Ullrich, I. Laptev, J. Verbeek, and H. Jégou, ''Image compression with product quantized masked image modeling,'' *Trans. Mach. Learn. Res.*, Jan. 2022.

[74] C. Zheng, L. T. Vuong, J. Cai, and D. Phung, ''MoVQ: Modulating quantized vectors for high-fidelity image generation,'' in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 23412–23425.

[75] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu, ''Vector-quantized image modeling with improved VQGAN,'' in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[76] S. Cao, Y. Yin, L. Huang, Y. Liu, X. Zhao, D. Zhao, and K. Huang, ''Efficient-VQGAN: Towards high-resolution image generation with efficient vision transformers,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 7334–7343.

[77] M. Huang, Z. Mao, Q. Wang, and Y. Zhang, ''Not all image regions matter: Masked vector quantization for autoregressive image generation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2002–2011.

SHOMA IWAI (Graduate Student Member, IEEE) received the B.E. and M.E. degrees from Tohoku University, Japan, in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree in communication engineering with the IIC-Laboratory. His current research interests include computer vision and image compression.

TOMO MIYAZAKI (Member, IEEE) received the B.E. degree from Yamagata University, in 2006, and the Ph.D. degree from Tohoku University, in 2011. He worked on a geographic information system with Hitachi Ltd., until 2013. He was a Postdoctoral Researcher and an Assistant Professor with Tohoku University, from 2013 to 2023. Since 2024, he has been an Associate Professor. His research interests include pattern recognition and image processing.

SHINICHIRO OMACHI (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in information engineering from Tohoku University, Japan, in 1988, 1990, and 1993, respectively. He was an Assistant Professor with the Education Center for Information Processing, Tohoku University, from 1993 to 1996. Since 1996, he has been affiliated with the Graduate School of Engineering, Tohoku University, where he is currently a Professor. From 2000 to 2001, he was a Visiting Associate Professor with Brown University. His current research interests include pattern recognition, computer vision, image processing, image coding, and parallel processing. He is a member of the Institute of Electronics, Information and Communication Engineers, and the Information Processing Society of Japan. He received the IAPR/ICDAR Best Paper Award, in 2007, the Best Paper Method Award of the 33rd Annual Conference of the GfKl, in 2010, the ICFHR Best Paper Award, in 2010, and the IEICE Best Paper Award, in 2012. He served as the Vice Chair for the IEEE Sendai Section, from 2020 to 2021. He served as the Editor-in-Chief for *IEICE Transactions on Information and Systems*, from 2013 to 2015.