

Exploring Distortion Prior with Latent Diffusion Models for Remote Sensing Image Compression

Junhui Li^{ID}, Jutao Li, Xingsong Hou^{ID}, and Huake Wang^{ID}

Abstract—Learning-based image compression algorithms typically focus on designing encoding and decoding networks and improving the accuracy of entropy model estimation to enhance the rate-distortion (RD) performance. However, few algorithms leverage the compression distortion prior from existing compression algorithms to improve RD performance. In this paper, we propose a latent diffusion model-based remote sensing image compression (LDM-RSIC) method, which aims to enhance the final decoding quality of RS images by utilizing the generated distortion prior from a LDM. Our approach consists of two stages. In Stage I, a self-encoder learns prior from the high-quality input image. In Stage II, the prior is generated through a LDM, conditioned on the decoded image of an existing learning-based image compression algorithm, to be used as auxiliary information for generating the texture-rich enhanced images. To better utilize the prior, a channel attention and gate-based dynamic feature attention module (DFAM) is embedded into a Transformer-based multi-scale enhancement network (MEN) for image enhancement. Extensive experimental results demonstrate the proposed LDM-RSIC outperforms existing state-of-the-art traditional and learning-based image compression algorithms in terms of both subjective perception and objective metrics. The code will be available at <https://github.com/mlkk518/LDM-RSIC>.

Index Terms—Image compression, latent diffusion models, remote sensing image, image enhancement.

I. INTRODUCTION

WITH the ongoing development of remote sensing (RS) technology, the volume of RS images is growing dramatically [1]. This mainly stems from the continuous upgrading of platforms such as satellites and airplanes, as well as the wide application of high-resolution sensors [2], [3]. Compression can help reduce the cost of data storage and transmission, and improve the efficiency of data processing and analysis, especially important in tasks that use high-resolution RS images for real-time monitoring. Moreover, with the wide application of RS images in various applications, there is an increasing demand for fast data acquisition, sharing, and processing. Consequently, the necessity to compress RS images is becoming increasingly imperative.

Conventional image compression methods like JPEG2000 [4], BPG [5], and VVC [6] have been pivotal in facilitating the storage and transmission of image data. However, these established standards suffer from two notable drawbacks [7],

Manuscript received xx, 2024; This work was supported by the National Natural Science Foundation of China under Grant 62272376. (*Corresponding authors:* Xingsong Hou.)

Junhui Li, Jutao Li, Xingsong Hou, and Huake Wang are with the School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: mlkkjh@stu.xjtu.edu.cn, houxus@mail.xjtu.edu.cn).

[8]. Firstly, the encoding or decoding processes necessitate a sequential block-by-block implementation in block-based hybrid codes, leading to undesired blocking or ringing artifacts in the decoded images. Secondly, the intricate interdependencies among hand-crafted modules make it challenging to jointly optimize the entire coding algorithm. Furthermore, the growing demand for high-resolution RS images and the emergence of diverse applications with specific requirements have led to the exploration of advanced compression methods.

With the development of deep learning technology, learning-based image compression algorithms have made great progress [9], [10], [11], [12], [13]. Particularly, Ballé *et al.* [10] first introduced additional side information in the form of a hyperprior entropy model to estimate a zero-mean Gaussian distribution, highly improving the rate-distortion (RD) performance. Furthermore, Cheng *et al.* [8] proposed the utilization of discretized Gaussian mixture likelihoods to parameterize the distributions of the entropy model, resulting in impressive RD performance. To further obtain higher RD performance and running speed, He *et al.* [11] proposed an uneven channel-conditional adaptive grouping method to improve the prediction accuracy of the entropy model, and developed the efficient learned image compression (ELIC) algorithm. However, although the above learning-based compression algorithms have achieved impressive RD performance in natural scenes, RS image compression has not made significant progress due to the rich texture, context, and spectral information in RS images [14], [15].

To overcome this obstacle, Zhang *et al.* [16] strengthened the network's feature extraction capabilities by introducing a multi-scale attention module. To improve the entropy model, they also added global priors and anchored-stripe attention. Pan *et al.* [2] employed generative adversarial networks (GANs) to separately decode image content and complex textures for effective low-bitrate RS image compression. Wang *et al.* [17] observed that the conventional method of acquiring satellite images only uses the downlink to send compressed data to ground stations. They proposed using the uplink to utilize historical ground station images as references for on-orbit compression, reducing redundancy in RS images and improving compression efficiency. Additionally, Xiang *et al.* [18] used discrete wavelet transform to separate image characteristics into high- and low-frequency components, then developed compression networks to accurately represent both types of characteristics for high compression efficiency.

While the aforementioned algorithms have demonstrated remarkable RD performance in RS image compression, deriving text-rich decoded images, especially at low bitrates, remains

challenging. Recently, latent diffusion models (LDMs) have garnered significant attention due to their powerful ability to balance complexity reduction and intricate detail preservation [19], [20], [21], [22], [23]. Unlike traditional DMs [24], [25], which typically operate directly in pixel space and require extensive computational resources for model optimization, LDMs can train models with limited computational resources while retaining high quality and flexibility [26]. For example, Chen *et al.* [21] deployed a DM in latent space (*i.e.*, LDM) to generate prior, which were then embedded into an enhanced network through a hierarchical integration module for image deblurring. Similarly, Corneanu *et al.* [22] performed forward and backward fusion steps using LDM, achieving impressive performance in terms of both image inpainting and running efficiency.

In this paper, we propose a novel LDM-based remote sensing image compression (LDM-RSIC) method. Specifically, our approach comprises two main stages: In Stage I, we develop a self-encoder to learn the compression distortion prior information from an existing image compression algorithm, ELIC. Prior information here refers to a compact representation of the residual or distortion between the original and compressed images, which helps the model understand and correct compression-induced detail loss. This prior encapsulates knowledge about the specific types of distortion commonly introduced during compression, particularly at low bitrates, such as loss of texture details or sharpness. By learning this prior, the model gains the ability to predict and mitigate these distortions during the decoding process. In Stage II, we generate this prior information using a LDM, conditioned on the decoded images. This generated prior information serves as auxiliary data and is fed into a multi-scale enhancement network (MEN) to enhance the quality of the decoded images. The key idea is that by utilizing the prior information, the MEN can better recover fine textures and structural details that may have been lost during the initial compression. Furthermore, a channel and gate-based dynamic feature attention module (DFAM) is embedded into the MEN for better integration of the prior information with the decoded images.

The primary contributions of this paper can be summarized as follows:

- We propose LDM-RSIC for RS image compression, leveraging the power of LDM to generate compression distortion prior, which is then utilized to enhance the image quality of the decoded images. The proposed LDM-based scheme can be adopted to improve the RD performance of both the learning-based and traditional image compression algorithms.
- We employ LDM and develop forward-backward fusion steps on the latent space to generate prior information instead of the pixel space for RS image compression. This approach not only saves on expensive training but also reduces inference time.
- A channel attention and gate-based DFAM is embedded in a Transformer-based MEN, facilitating dynamic fusion between the features of decoded images and the prior information.

- Extensive experiments conducted on two RS image datasets demonstrate the superior performance of LDM-RSIC compared to state-of-the-art traditional and learning-based image compression algorithms.

The remainder of the paper is structured as follows. Section II provides a review of related work on learning-based image compression and DMs. Section III elaborates on the proposed LDM-RSIC. Section IV presents the experimental results and analysis conducted to evaluate the performance of the proposed method. Finally, Section V concludes the paper.

II. RELATED WORK

In this section, we present related work from two perspectives. Firstly, we review the realm of learning-based image compression, an emerging field that has garnered significant attention in recent years. Secondly, we explore the DMs, which serve as a key source of inspiration for the proposed LDM-RSIC.

A. Learning-based Image Compression

In recent years, the rapid advancement of deep learning has promoted the development of numerous learning-based image compression techniques [16], [27], [11], [2]. These advancements build upon the pioneering work by Ballé *et al.* [10], who introduced a hyperprior entropy model to estimate zero-mean Gaussian distributions, facilitating compact latent representation within an end-to-end variational autoencoder-based framework [28].

When developing learning-based image compression methods, there are typically two key considerations. First, the redundancy coefficients among latent representations are reduced by improving the encoder network to save coding streams. Second, more accurate entropy models are designed to accurately estimate the probability distribution of these coefficients, thus better controlling the bitrate required for potential representation coding [29].

In natural image compression, many efforts have been dedicated to achieving compact representation through the refinement of encoder and decoder networks [16], [30], [31], [32]. For instance, Gao *et al.* [32] proposed a back projection scheme with attention and multi-scale feature fusion. Tang *et al.* [30] employed asymmetric convolutional neural networks and attention mechanisms to enhance RD performance. Also, accurate entropy model development has garnered wide attention [10], [33], [34], [27], [8], [11], [35]. Specifically, Cheng *et al.* [8] introduced discretized Gaussian mixture likelihoods for entropy model parameterization. Qian *et al.* [33] proposed a Transformer-based entropy model by leveraging the Transformer's long-range dependency capabilities. Moreover, He *et al.* [11] developed an uneven channel-conditional adaptive grouping scheme to enhance coding performance without compromising speed. Liu *et al.* [36] incorporated the local modeling ability of CNN and the non-local modeling ability of Transformers and developed a Transformer-CNN Mixture (TCM) block, which was adopted to improve the RD performance.

In RS image compression, several studies focused on exploring competitive network structures and image transformations [2], [18], [7], [16]. For example, Pan *et al.* [2] utilized GANs to independently decode image content and detailed textures, thereby enhancing compression performance at low bitrates. Xiang *et al.* [18] leveraged discrete wavelet transforms to enhance the representation of high and low-frequency features. Li *et al.* [3] introduced Transformer and patch-based local attention modules to develop a competitive encoder network and entropy model for object-fidelity RS image compression. Pan *et al.* [7] proposed a hybrid attention network to improve entropy model prediction accuracy. Zhang *et al.* [16] employed a multi-scale attention module and global priors to enhance feature extraction and improve the entropy model. In contrast to these approaches, our objective is to employ conditional LDM to generate compression distortion prior induced by one of the existing compression algorithms. Subsequently, this prior information is utilized to enhance the RD performance of the compression algorithm.

B. Diffusion Models (DMs)

DMs [37], [38], [37], as probabilistic generative models, utilize parameterized Markov chain to optimize the lower variational bound on the likelihood function. This enables the construction of desired data samples from Gaussian noise via a stochastic iterative denoising process. Recently, DMs have become increasingly influential in image restoration tasks, such as image super-resolution [24], inpainting [25], and deblurring [39]. However, since these models typically operate directly in pixel space, optimizing the most powerful DMs often requires hundreds of GPU days, and the cost of inference is also high due to sequential evaluation [26].

To train DMs on limited computational resources while maintaining their quality and flexibility, Rombach *et al.* [26] innovatively applied them to the latent space of powerful pretrained autoencoders, thus developing the LDM. Compared to previous DM-based methods, LDM achieves a near-optimal balance between reducing complexity and preserving details for the first time, greatly enhancing visual fidelity. Subsequently, a series of LDM-based studies have been reported in a wide range of applications [20], [21], [22]. For example, in [21], the authors leveraged LDM to generate prior features, which were then fused to an enhancement network by a hierarchical integration module for image deblurring. Besides, in [22], the authors resorted to the powerful ability of LDM and developed an impressive LDM-based image inpainting algorithm.

III. METHODOLOGY

In this section, the proposed LDM-RSIC and its two novel designs will be introduced in detail. We will first begin with an overview of LDM-RSIC. Subsequently, two novel designs, namely, prior information learning and LDM-based prior information generation, will be presented.

A. Overview of LDM-RSIC

The main framework of the proposed LDM-RSIC is depicted in Fig. 1, where “Conv | k3 | s2” refers to the convolution layer with a kernel size of 3×3 and stride 2. By default, the stride value is set to 1. This framework primarily comprises three components: the compressor, multi-scale enhancement network (MEN), and latent diffusion module (LDM).

Specifically, the compressor refers to an existing image compression algorithm, here the competitive algorithm ELIC is employed as the compressor. Given an input image $\mathbf{X} \in \mathbb{R}^{W \times H \times 3}$, the encoder g_a aims to extract the latent features \mathbf{Y} in a compact manner. These features \mathbf{Y} are then quantized by a quantification Q to attain $\hat{\mathbf{Y}}$. By encoding $\hat{\mathbf{Y}}$ with a hyper entropy coder, the image can be compressed into a data stream. On the decompression side, the reconstructed image can be obtained using $\hat{\mathbf{Y}}$ and the decoder network g_s . The above process can be formulated as

$$\begin{aligned}\mathbf{Y} &= g_a(\mathbf{X}; \phi), \\ \hat{\mathbf{Y}} &= Q(\mathbf{Y}), \\ \tilde{\mathbf{X}} &= g_s(\hat{\mathbf{Y}}; \varphi),\end{aligned}\quad (1)$$

where the encoder g_a and decoder g_s represent neural networks developed in [11], with ϕ and φ denoting their network parameters, respectively. $Q(\cdot)$ denotes a quantization operation. To encode the latent $\hat{\mathbf{Y}}$, hyper networks h_a and h_s are typically developed to obtain the probability model, which is commonly used for arithmetic coding. For achieving different compression rates, the compression networks are often optimized using the loss function defined as

$$\mathcal{L}_{\text{com}} = R + \lambda D(\mathbf{X}, g_s(\hat{\mathbf{Y}})), \quad (2)$$

where the rate R represents the entropy calculation function of the quantized latent representation of the compressor, and $D(\cdot)$ denotes the mean squared error (MSE). λ is a hyperparameter used to control the balance between the rate and the quality of the decoded images, where a higher value of λ indicates better quality of the decoded image and higher bitrates.

Over the past few years, numerous studies have focused on designing the network structures of g_a , g_s , h_a , and h_s . However, it seems challenging to further improve RD performance solely through network design. Considering the impressive ability of LDMs to generate rich information conditioned on known knowledge, we aim to employ the LDM to generate prior information, which is then embedded into MEN to enhance the quality of the decoded images, thus improving the RD performance of the compressor. This process can be achieved through two stages: prior information learning and prior information generation. Next, we will illustrate the two stages in detail.

B. Stage I: Prior Information Learning

In Stage I, as illustrated in Fig. 1(b), both the input image \mathbf{X} and the decoded image $\tilde{\mathbf{X}}$ are fused through the latent representation module (LRM) to obtain latent representation features \mathbf{F} . The features \mathbf{F} , enriched with high-quality prior information, are then utilized to assist the MEN in reconstructing the enhanced image $\hat{\mathbf{X}}$.

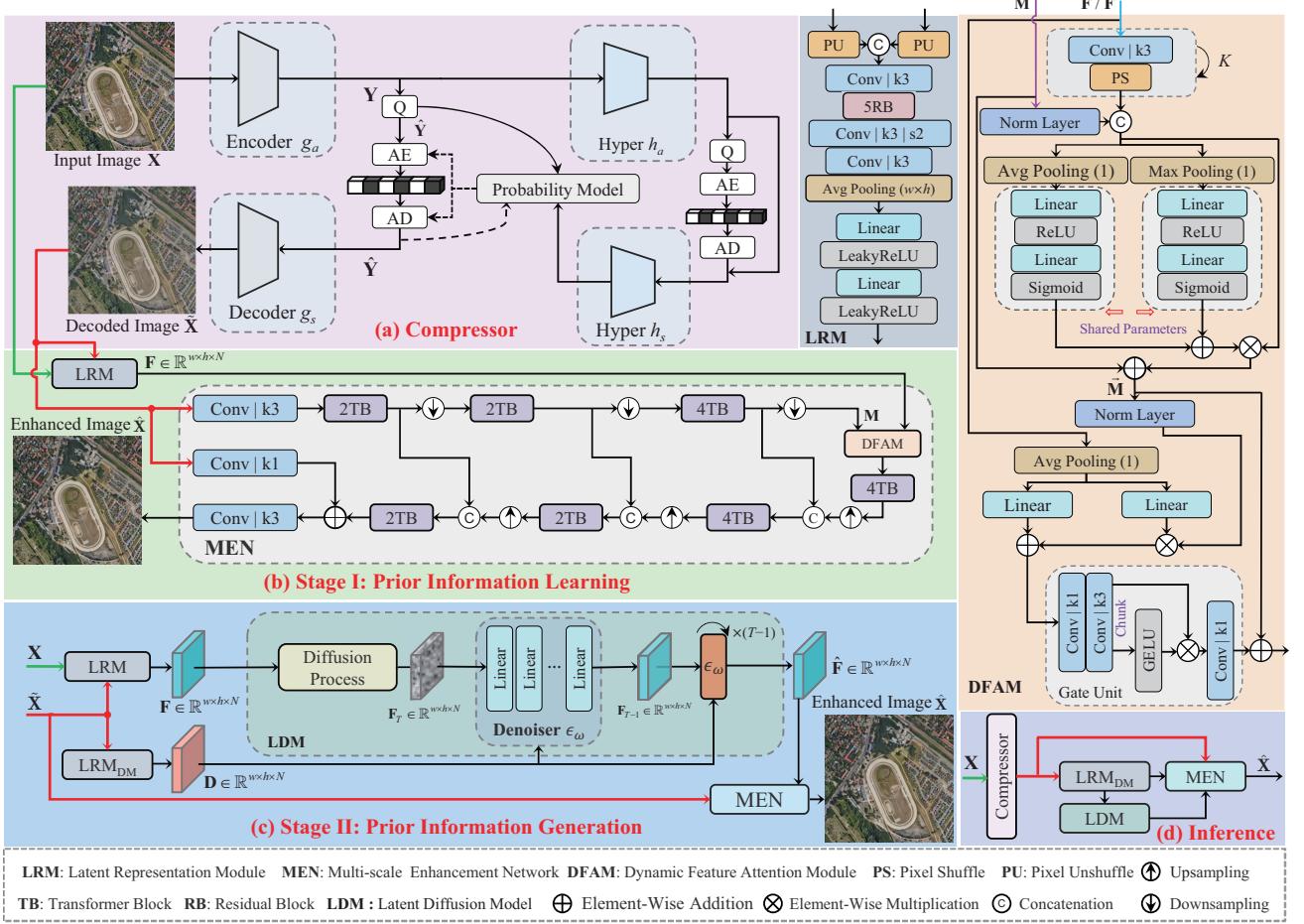


Fig. 1. Overview of the proposed LDM-RSIC, which comprises the compressor, MEN, and LDM. The compressor utilizes the competitive image compression algorithm ELIC [11]. “2TB” indicates two stacked Transformer blocks, and “5RB” denotes five serially connected residual blocks. “AE” and “AD” refer to the arithmetic encoder and decoder, respectively. Stage I aims to learn the prior information \mathbf{F} , and Stage II focuses on employing LDM to generate the prior features $\hat{\mathbf{F}}$ to replace \mathbf{F} .

1) *Latent Representation Module (LRM)*: To be specific, as depicted in the LRM component of Fig. 1, the input image and decoded image undergo a pixel unshuffle (PU) operation to downsample the features. Subsequently, these downsampled features are concatenated and fed into several convolutional and residual block (RB) layers, followed by an average pooling layer with an output size of $w \times h$ to reduce the dimensionality. Afterward, two linear layers with LeakyReLU activation functions are employed to mixture the features. This process yields the latent representation given as

$$\mathbf{F} = \text{LRM}(\tilde{\mathbf{X}}, \mathbf{X}), \quad (3)$$

where the resolution of \mathbf{F} is $w \times h \times N$. Importantly, the token number $w \times h \times N$ remains a constant much smaller than $W \times H \times 3$. Consequently, the computational burden of the subsequent LDM is effectively reduced. In addition, it should be noted that each “Conv” in LRM of Fig. 1 is followed by a LeakyReLU activation function.

2) *Multi-Scale Enhance Network (MEN)*: Considering the impressive performance of multi-scale Transformer-based networks in various tasks [40], [41], [21], MEN is developed based on Transformer blocks (TBs) for the quality improve-

ment of the decoded images. Furthermore, to better utilize the latent features \mathbf{F} , a channel attention and gate-based dynamic feature attention module (DFAM) is embedded in MEN.

In the development of DFAM, we first upsample the features \mathbf{F} through K stacked layers comprising a convolutional layer followed by a pixel shuffle (PS) layer. Subsequently, the MEN’s deep features \mathbf{M} , after a layer normalization operation, are concatenated with the upsampled features along the channel dimension. As different channels hold varying importance for the final reconstruction, channel attention is introduced to dynamically weigh the concatenated channels. Thus, as depicted in Fig. 1, the output features $\vec{\mathbf{M}}$ can be expressed as

$$\begin{aligned} \mathbf{Z} &= \text{Concat}(\text{UP}(\mathbf{F}), \text{Norm}(\mathbf{M})), \\ \vec{\mathbf{M}} &= (\text{FC}(\text{AP}(\mathbf{Z})) + \text{FC}(\text{MP}(\mathbf{Z}))) \otimes \mathbf{Z} + \mathbf{M}, \end{aligned} \quad (4)$$

where $\text{UP}(\cdot)$ denotes the upsampling operation function described above; $\text{FC}(\cdot)$, $\text{AP}(\cdot)$, and $\text{MP}(\cdot)$ represent the full connection layer, average pooling, and max pooling, respectively. \otimes indicates element-wise multiplication.

Additionally, since not all learned latent features \mathbf{F} contribute equally to the improvement of the reconstruction, we incorporate a gate adjuster to further reweight the obtained

features $\vec{\mathbf{M}}$. Specifically, before further processing, the prior information \mathbf{F} adjusts the features $\vec{\mathbf{M}}$ through scaling and shifting operations, followed by a gate unit (GU) to selectively amplify or suppress certain features based on their relevance. Therefore, the output of DFAM can be derived as

$$\vec{\mathbf{M}} = \text{GU}(\text{Norm}(\vec{\mathbf{M}}) \otimes \text{LL}(\text{AP}(\mathbf{F})) + \text{LL}(\text{AP}(\mathbf{F}))) + \vec{\mathbf{M}}, \quad (5)$$

where $\text{LL}(\cdot)$ refers to a linear layer. This integration allows the model to adjust $\vec{\mathbf{M}}$ based on the prior information \mathbf{F} , which enhances the flexibility and adaptability of the model.

3) *Prior Learning Loss*: To obtain the latent features \mathbf{F} , the pretrained model parameters of ELIC are frozen, and MEN and LRM are jointly optimized using the L1 norm given as

$$\mathcal{L}_{\text{learn}}(\Omega_1) = \|\text{MEN}(\tilde{\mathbf{X}}, \text{LRM}(\tilde{\mathbf{X}}, \mathbf{X})) - \mathbf{X}\|_1, \quad (6)$$

where $\tilde{\mathbf{X}}$ is the decoded image of ELIC with a specific value of λ . Ω_1 refers to the trained network parameters. Thus, the process of stage I can be summarized in Algorithm 1.

Algorithm 1: Process of stage I

```

1 Input:  $\mathbf{X}, \lambda$ 
2 Output:  $\Omega_1, \mathbf{F}$ 
3: Prior Information Learning:
4:   Freeze the pretrained ELIC.
5:   for epoch  $\leftarrow 1$  to epochs1 do
6:      $\tilde{\mathbf{X}} = \text{ELIC}(\mathbf{X}, \lambda)$ ,
7:     // Learned prior features:
8:      $\mathbf{F} = \text{LRM}(\tilde{\mathbf{X}}, \mathbf{X})$ ,
9:     // Compute prior learning loss:
10:     $\mathcal{L}_{\text{learn}}(\Omega_1) = \|\text{MEN}(\tilde{\mathbf{X}}, \mathbf{F}) - \mathbf{X}\|_1$ ,
11:    // Update  $\Omega_1$  using gradient descent:
12:     $\Omega_1 \leftarrow \Omega_1 - \alpha_1 \nabla \mathcal{L}_{\text{learn}}(\Omega_1)$ ,
13:   end for
14: Return:  $\Omega_1$ 

```

C. Stage II: Prior Information Generation

In Stage II, the LDM is trained to generate the prior features $\hat{\mathbf{F}}$, which replace the features \mathbf{F} of MEN for higher enhanced image quality. Concretely, the LDM is developed based on conditional DDPM [42], [43], [44]. The LDM involves a forward diffusion process and a reverse denoising process, as depicted in Fig. 1(c).

1) *Diffusion Process*: Given an input image \mathbf{X} , we first employ the LRM trained in Stage I to generate the corresponding prior features $\mathbf{F} \in \mathbb{R}^{w \times h \times N}$. We take \mathbf{F} as the starting point of the forward Markov process and gradually add Gaussian noise to it over T iterations as follows:

$$q(\mathbf{F}_{1:T} | \mathbf{F}_0) = \prod_{t=1}^T q(\mathbf{F}_t | \mathbf{F}_{t-1}), \quad (7)$$

$$q(\mathbf{F}_t | \mathbf{F}_{t-1}) = \mathcal{N}(\mathbf{F}_t; \sqrt{1-\eta_t} \mathbf{F}_{t-1}, \eta_t \mathbf{I}),$$

where \mathbf{F}_t represents the noisy features at the t -th step and $\mathbf{F}_0 = \mathbf{F}$; $\eta_{1:T} \in (0, 1)$ are hyperparameters that control the variance of the noise; \mathcal{N} denotes the Gaussian distribution. Based on the iterative derivation presented in [45], we can reformulate Eq. (7) as

$$q(\mathbf{F}_T | \mathbf{F}_0) = \mathcal{N}(\mathbf{F}_T; \sqrt{\gamma_T} \mathbf{F}_0, (1 - \gamma_T) \mathbf{I}), \quad (8)$$

where $\bar{\gamma}_t = \prod_{t=1}^T \gamma_t$ and $\gamma_t = 1 - \eta_t$.

2) *Reverse Process*: Here, our objective is to generate the prior features from a pure Gaussian distribution. The reverse process is a T -step Markov chain that runs backward from \mathbf{F}_T to \mathbf{F}_0 . Specifically, for the reverse step from \mathbf{F}_t to \mathbf{F}_{t-1} , we use the posterior distribution given as

$$p(\mathbf{F}_{t-1} | \mathbf{F}_t, \mathbf{F}_0) = \mathcal{N}\left(\mathbf{F}_{t-1}; \mu_t(\mathbf{F}_t, \mathbf{F}_0), \frac{1 - \bar{\gamma}_{t-1}}{1 - \bar{\gamma}_t} \eta_t \mathbf{I}\right), \quad (9)$$

$$\text{s.t., } \mu_t(\mathbf{F}_t, \mathbf{F}_0) = \frac{1}{\sqrt{\gamma_t}} \left(\mathbf{F}_t - \frac{1 - \gamma_t}{\sqrt{1 - \bar{\gamma}_t}} \epsilon \right),$$

where ϵ represents the noise in \mathbf{F}_t , and is the only uncertain variable. Following previous works [41], [21], we adopt a neural network, termed as denoiser (ϵ_ω), to estimate the noise ϵ for each step. Since the LDM operates in the latent space, we utilize another latent encoder, denoted as LRM_{DM} , which has the same structure as the network after LRM removes the single PU input branch. LRM_{DM} compresses the decoded image $\tilde{\mathbf{X}}$ into latent space to get the condition latent features $\mathbf{D} \in \mathbb{R}^{w \times h \times N}$. The denoising network predicts the noise conditioned on \mathbf{F}_t and \mathbf{D} , i.e., $\epsilon_\omega(\mathbf{F}_t, \mathbf{D}, t)$. With the substitution of ϵ_ω in Eq. (9) and set the variance to $(1 - \gamma_t)$, thus we can derive

$$\mathbf{F}_{t-1} = \frac{1}{\sqrt{\gamma_t}} (\mathbf{F}_t - \frac{1 - \gamma_t}{\sqrt{1 - \bar{\gamma}_t}} \epsilon_\omega(\mathbf{F}_t, \mathbf{D}, t)) + \sqrt{1 - \gamma_t} \epsilon_t, \quad (10)$$

where $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$. By iteratively sampling \mathbf{F}_t using Eq. (10) T times, we can generate the predicted prior features $\hat{\mathbf{F}} \in \mathbb{R}^{w \times h \times N}$, as depicted in Fig. 1(c). The predicted prior features are then used to guide MEN. It should be noted that as the distribution of the latent features $\mathbf{F} \in \mathbb{R}^{w \times h \times N}$ is much simpler than that of the input image $\mathbf{X} \in \mathbb{R}^{W \times H \times 3}$, the prior features $\hat{\mathbf{F}}$ can be generated with a small number of iterations.

3) *Prior Generation Loss*: In this stage, the models of ELIC and LRM from Stage I are frozen, and the pretrained network parameters of LRM and MEN are respectively used to initialize the networks of LRM_{DM} and MEN for Stage II. Training LDM involves training the denoising network ϵ_ω . As adopted in the studies [26], [46], we train the model by optimizing the network parameters ω of ϵ_ω , which can be formulated as

$$\nabla_\omega \|\epsilon - \epsilon_\omega(\sqrt{\bar{\gamma}_t} \mathbf{F} + \sqrt{1 - \bar{\gamma}_t} \epsilon, \mathbf{D}, t)\|_2^2, \quad (11)$$

where \mathbf{F} and \mathbf{D} refer to the learned prior features and condition latent features derived above; $t \in [1, T]$ is a random time step, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ denotes sampled Gaussian noise.

For each training iteration, we use the prior features \mathbf{F} to generate the noise sample \mathbf{F}_T according to Eq. (8). Given that the time-step T is relatively small in latent space, we then run the complete T iteration reverse processes (i.e., Eq. (10)) to generate the predicted prior features $\hat{\mathbf{F}}$, which can be derived by

$$\hat{\mathbf{F}} = \text{LDM}(\text{LRM}_{\text{DM}}(\tilde{\mathbf{X}})). \quad (12)$$

The obtained features $\hat{\mathbf{F}}$ are then used to provide prior information for MEN. Thereafter, the loss function can be formulated as

$$\mathcal{L}_{\text{gen}}(\Omega_2) = \|\text{MEN}(\tilde{\mathbf{X}}, \hat{\mathbf{F}}) - \mathbf{X}\|_1 + \|\hat{\mathbf{F}} - \mathbf{F}\|_1, \quad (13)$$

where the first term on the right side of the equation is the quality fidelity item, and the second term is the diffusion loss.

Algorithm 2: Process of stage II

```

1 Input:  $\mathbf{X}, \Omega_1, \mathbf{F}, \lambda$ 
2 Output:  $\Omega_2$ 
3 Prior Information Generation:
4   Freeze the pretrained ELIC and LRM;
5   Initialize MEN and  $\text{LRM}_{\text{DM}}$  with  $\Omega_1$ .
6   for epoch  $\leftarrow 1$  to epochs2 do
7     // Learned prior features:
8      $\mathbf{F} = \text{LRM}(\tilde{\mathbf{X}}, \mathbf{X})$ ,
9     // Generated prior features:
10     $\hat{\mathbf{F}} = \text{LDM}(\text{LRM}_{\text{DM}}(\tilde{\mathbf{X}}))$ ,
11    // Compute prior generation loss:
12     $\mathcal{L}_{\text{gen}}(\Omega_2) = \|\text{MEN}(\tilde{\mathbf{X}}, \hat{\mathbf{F}}) - \mathbf{X}\|_1 + \|\hat{\mathbf{F}} - \mathbf{F}\|_1$ ,
13    // Update  $\Omega_2$  using gradient descent:
14     $\Omega_2 \leftarrow \Omega_2 - \alpha_2 \nabla \mathcal{L}_{\text{gen}}(\Omega_2)$ ,
15  end for
16 Return:  $\Omega_2$ 


---


Input:  $\mathbf{X}, \Omega_2, \lambda$ 
Output:  $\hat{\mathbf{X}}$ 
1 Inference:
2 Load pretrained ELIC.
3  $\tilde{\mathbf{X}} = \text{ELIC}(\mathbf{X}, \lambda)$ ,
4 Load  $\Omega_2$  for  $\text{LRM}_{\text{DM}}$ , LDM, and MEN.
5  $\hat{\mathbf{F}} = \text{LDM}(\text{LRM}_{\text{DM}}(\tilde{\mathbf{X}}))$ 
6  $\hat{\mathbf{X}} = \text{MEN}(\tilde{\mathbf{X}}, \hat{\mathbf{F}}; \Omega_2)$ 
7 Return:  $\hat{\mathbf{X}}$ 

```

4) *Inference*: As depicted in Fig. 1(d), after obtaining the trained model, the input image \mathbf{X} undergoes compression using the compressor, ELIC. Subsequently, the compressed image $\tilde{\mathbf{X}}$ is fed into Eq. (12) to generate the prior features $\hat{\mathbf{F}}$. Following this, MEN utilizes $\hat{\mathbf{F}}$ to produce the enhanced image $\hat{\mathbf{X}}$ with rich texture details. This process can be expressed as

$$\begin{aligned} \tilde{\mathbf{X}} &= \text{ELIC}(\mathbf{X}, \lambda), \\ \hat{\mathbf{X}} &= \text{MEN}(\tilde{\mathbf{X}}, \hat{\mathbf{F}}; \Omega_2), \end{aligned} \quad (14)$$

where $\hat{\mathbf{X}}$ refers to the enhanced image. Thereafter, the process of stage II and inference can be summarized in Algorithm 2.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Settings

1) *Datasets*: To evaluate the performance of the proposed LDM-RSIC, we conducted experiments using two RS image datasets: DOTA [47] and UC-Merced (UC-M) [48], as adopted in [3], [7]. The DOTA dataset consists of 2,806 images, each with pixel resolutions ranging from 800×800 to 4000×4000 , containing objects of various scales, orientations, and shapes. The UC-M dataset contains 21 classes of RS scenes and a total of 2,100 images, each with a resolution of 256×256 .

During training, we use the training dataset of DOTA for model training and employ several augmentation strategies, such as random horizontal flip, random vertical flip, and random crop. To evaluate the performance of the proposed method, we construct the testing set as follows: We randomly select 100 images from the testing dataset of DOTA, and each

of these images is then cropped to a resolution of 256×256 . Additionally, we select 20% of the images from each category in the UC-M dataset.

2) *Implementation Details*: The experiments are conducted on an Intel Silver 4214R CPU with 2.40GHz and one NVIDIA GeForce RTX 3090 Ti GPU, using the PyTorch 1.13.1 framework with CUDA 11.7. ELIC [11] is employed to compress the images of the training dataset. Specifically, five trained models with parameters $\lambda \in \{4, 8, 32, 100, 450\} \times 10^{-4}$ are used to separately generate the decoded images, which serve as input for training the developed model. Since different values of λ correspond to varying levels of compression distortion in ELIC, we use the decoded image under a specific λ to train a LDM-RSIC model. The values of w, h, N , and K are set to 4, 4, 256, and 3, respectively. Besides, we utilize the widely used Adam optimizer [49] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for model optimization. In Stage I, the initial learning rate α_1 is set to 1×10^{-4} , and the total number of iterations is 300K. After 92K iterations, the model is trained with the initial learning rate gradually reduced to 1×10^{-6} using cosine annealing [50]. Early stopping is applied to prevent overfitting. In Stage II, the total epochs are set to 400K, with an initial learning rate of $\alpha_2 = 1 \times 10^{-4}$ for training the LRM_{DM} and LDM before 200K iterations. Subsequently, the learning rate α_2 is reduced to 0.1 times every 80K iterations to jointly finetune the MEN, LRM_{DM} , and LDM. The batch size is set to 4 during the two stages.

3) *Evaluation Metrics*: In this paper, we use bits per pixel (bpp) to evaluate coding bitrates, while peak signal-to-noise ratio (PSNR), multi-scale structural similarity (MS-SSIM), and learned perceptual image patch similarity (LPIPS) [51] serve as evaluation metrics. PSNR and MS-SSIM focus on numerical comparisons and structural similarity in images, while LPIPS emphasizes perceptual evaluation by leveraging deep learning-based models to extract high-level perceptual features.

B. Quantitative Comparison

To demonstrate the effectiveness of the proposed LDM-RSIC, we compare it with three traditional image compression standards, including JPEG2000 [4], BPG [5] as well as VVC (YUV 444) [6], and the recent learning-based image compression algorithms, including Entrorformer [33], STF [53], ELIC [11], LIC-TCM [36], CDC ($\rho = 0.0$) [54], CDC ($\rho = 0.9$) [54], and WeConvne [55]. In addition, we further replace the learning-based compressor (*i.e.*, ELIC) in LDM-RSIC with the traditional algorithm JPEG2000, termed as JPEG2000*. Note that for DCD, the step number suggested by the authors was used in the experiments.

Figs. 2 (a) and (b) depict the RD curves of the proposed LDM-RSIC and the comparison algorithms on the testing sets of DOTA and UC-M, respectively. The results show that the overall performance of LDM-RSIC outperforms all other methods, including traditional and learning-based algorithms, except CDC ($\rho = 0.9$) on both datasets. We employ PSNR, MS-SSIM, and LPIPS metrics to measure the distortion of the decoded images. On both datasets, our RD curves outperform

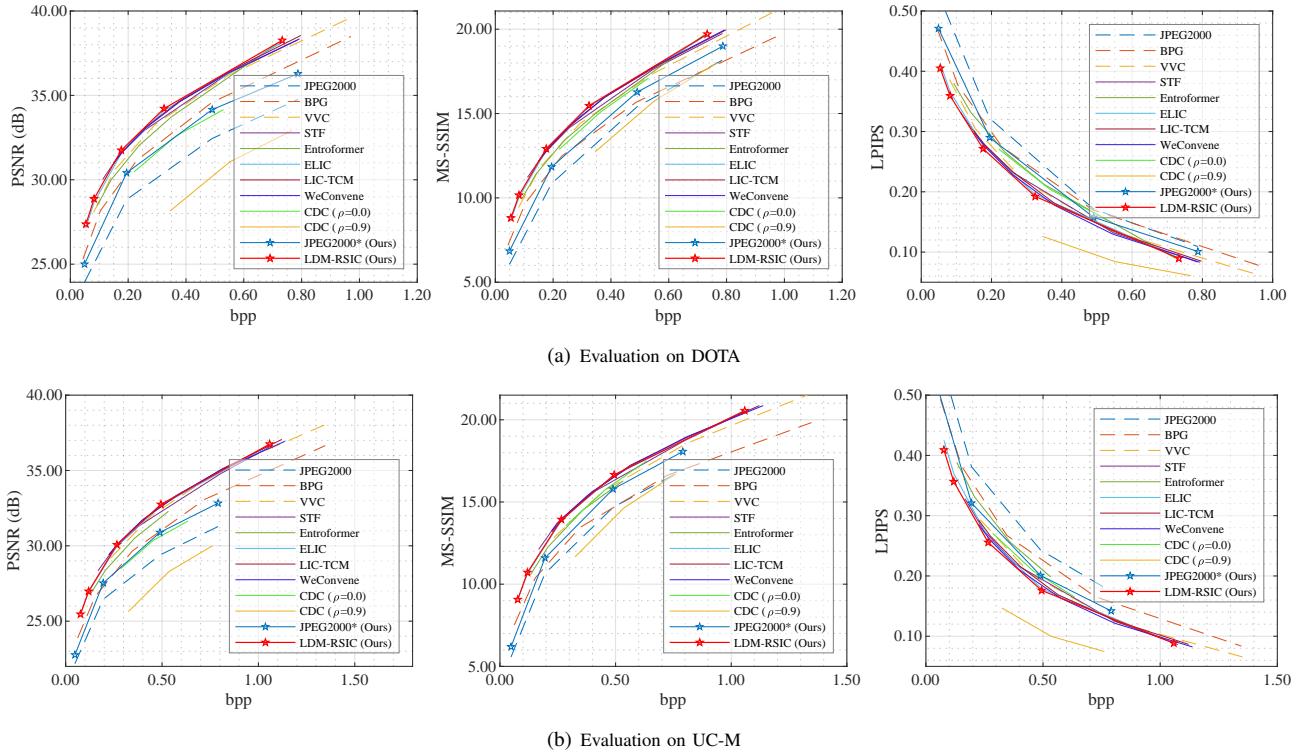


Fig. 2. Performance evaluation on the testing sets of DOTA and UC-M in terms of PSNR, MS-SSIM, and LPIPS. LDM-RSIC and JPEG2000* refer to algorithms designed using ELIC and JPEG2000 as the compressor in Fig. 1, respectively.

the comparison algorithm almost overall in terms of PSNR and MS-SSIM metrics, which indicates that LDM-RSIC achieves higher compression efficiency.

Furthermore, while CDC ($\rho = 0.9$) shows competitive performance in perceptual quality (as reflected by low LPIPS values), it struggles to achieve competitive PSNR values on both datasets. This suggests that CDC ($\rho = 0.9$) introduces synthetic, fake textures into the decoded images. In contrast, the proposed LDM-RSIC maintains high PSNR and achieves impressive LPIPS scores, demonstrating its balanced performance in both compression and perceptual quality.

Additionally, it is worth highlighting that JPEG2000*, when combined with the proposed scheme, delivers performance comparable to BPG, further showcasing the effectiveness of our method.

C. Qualitative Comparison

To visualize the performance of these image compression algorithms, Figs. 3 and 4 depict the decoded images “P0253”, “P0006”, “tenniscourt91”, and “intersection97” from the testing sets of DOTA and UC-M. At low bitrates, as illustrated in the image “P0253” of Fig. 3, artifacts and artifacts from traditional algorithms like JPEG2000 and BPG significantly affect the quality of the decoded images, particularly at low bitrates. It is worth noting the comparison with the state-of-the-art conventional compression algorithm, VVC. Although the bpp of VVC is slightly higher than that of LDM-RSIC, there is a significant difference in the PSNR values of the two, with VVC being 0.89 dB lower than LDM-RSIC. In addition,

it is clear that VVC is not able to adequately reconstruct the gridlines, whereas LDM-RSIC can clearly retain the edge information as well as have better contouring, demonstrating its superior performance.

When compared with learning-based algorithms, LDM-RSIC is observed to recover more details even at lower bitrates. For example, although the bpp of STF and LIC-TCM is 1.64 and 2.38 times higher than that of LDM-RSIC, respectively, LDM-RSIC demonstrates better recovery of the global structure of the images. Furthermore, at high bitrates, as shown in the image “P0006” of Fig. 3, one can see that LDM-RSIC obtains 1.40 dB PSNR gains than VVC even with fewer bitrate, and the edge details and structure information of LDM-RSIC are more impressive than VVC. Compared to the state-of-the-art learning-based image compression algorithm LIC-TCM, it can be observed that LDM-RSIC derives a 1.15 dB PSNR increase, and the edge information has been better preserved with a lower bitrate.

Moreover, as depicted in Fig. 4, despite VVC having a 11.55% higher bpp than LDM-RSIC, it fails to decode the structure of the windows. Notably, even though LIC-TCM has a bpp 2.07 times higher than LDM-RSIC, it still struggles to adequately recover the window structure. In the image “intersection97”, other methods struggle to clearly recover the red vehicles even at higher bpp. Although LIC-TCM exhibits higher PSNR and MS-SSIM scores than LDM-RSIC with higher bitrate, the road and vehicles appear more blurred. Therefore, we can safely demonstrate that the proposed LDM-RSIC presents a superior ability to decode texture-rich images

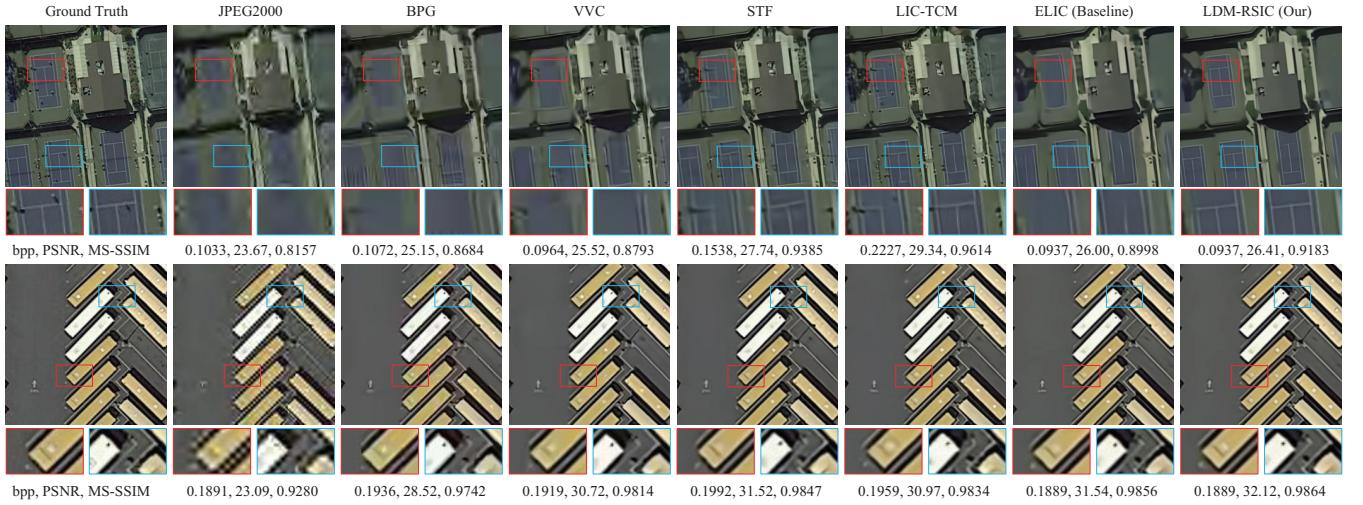


Fig. 3. Compressed images by several compression algorithms on the testing images “P0253” and “P0006” of the DOTA testing set.



Fig. 4. Compressed images by several compression algorithms on the images “tenniscourt91” and “intersection97” of the UC-M testing set.

compared to traditional and learning-based competitive image compression algorithms.

Additionally, Fig. 5 provides visual comparisons of the decoded images from CDC ($\rho = 0.9$), WeConvene, and the proposed LDM-RSIC. The results show that, despite consuming fewer bpp and achieving lower PSNR values, LDM-RSIC recovers significantly more edge details. Notably, although CDC ($\rho=0.9$) uses a higher bpp, both its PSNR and perceptual quality are inferior to those of LDM-RSIC. This strongly underscores the superior capability of the proposed method in enhancing perceptual quality.

D. Prior Information Analysis

Here, we focus on an analysis of the prior information of the developed LDM-RSIC and JPEG2000* from the two stages, namely the learned prior information in Stage I and the generated prior information in Stage II, to provide insights for future research directions. As outlined in Section III, Stage I is designed to learn the compression distortion prior conditioned

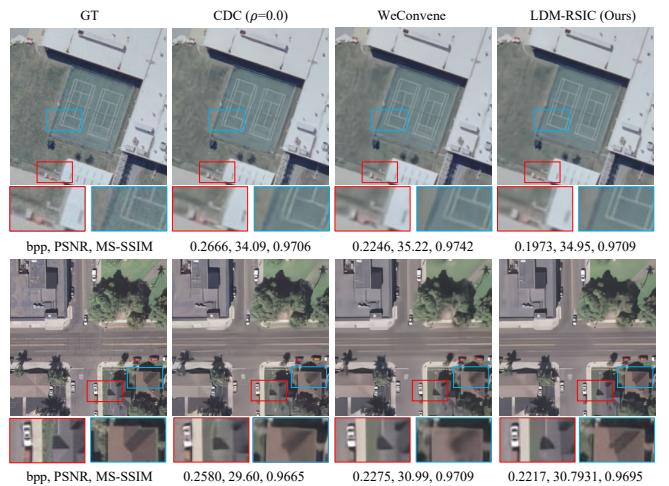


Fig. 5. Visualization of the decoded images “tenniscourt84” and “intersection94” using state-of-the-art compression methods and the proposed LDM-RSIC.

on both the ground truth and the decoded image, while Stage II focuses on leveraging LDM to generate the prior solely conditioned on the decoded image. The closer the generated prior in Stage II is to the prior learned in Stage I, the more enhanced results we can obtain.

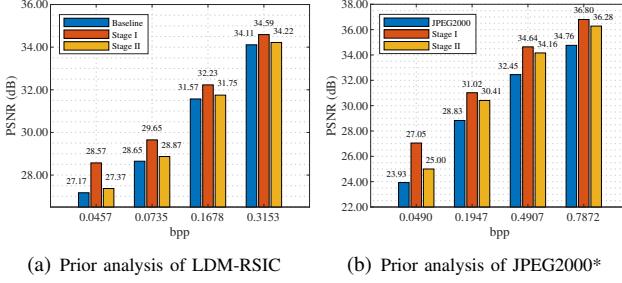


Fig. 6. PSNR performance of the proposed LDM-RSIC and JPEG2000* on Stages I and II with different values of bpp on the DOTA testing set.

We refer to ELIC, the learning-based compression algorithm utilized in the proposed LDM-RSIC, as the baseline. Figs. 6 (a) and (b) illustrate the PSNR of the proposed LDM-RSIC and JPEG2000* in Stages I and II with varying values of bpp. Take Fig. 6(a) as an example, it is evident that, under each bpp, the PSNR values of Stage I exhibit the highest results. Although the PSNR values of Stage II are higher than the baseline, the gap between Stages I and II cannot be considered negligible. For instance, under $\text{bpp} = 0.0457$, the PSNR values of the baseline, Stage I, and Stage II are 27.17 dB, 28.57 dB, and 27.37 dB, respectively. This indicates that the gap between the two stages is as high as 1.20 dB, suggesting that LDM still has great potential to improve the generated prior information and narrow this gap. Fig. 6(b) shows the same result. Hence, exploring more competitive LDM techniques is expected to further enhance the RD performance of the proposed LDM-RSIC and JPEG2000*.

E. Ablation Studies

In this section, we investigate the effects of different designs of the proposed method. We conduct all experiments on the DOTA testing set. Here, we first explore the value of the diffusion step T , then verify the validity of the LDM-based generated prior, and finally confirm the effectiveness of the developed LDM-based scheme in improving the compression efficiency of learning-based and traditional image compression algorithms.

1) *Diffusion Step*: To explore the effect of the diffusion step T in the LDM on the performance of the proposed LDM-RSIC, we vary the total number of iteration steps of the LDM-RSIC and adjust the parameter η_t in Eq. (8) to ensure that the \mathbf{F}_T becomes Gaussian noise after the diffusion process, where $\mathbf{F}_T \sim \mathcal{N}(0, \mathbf{I})$. Fig. 7 illustrates the average PSNR of the proposed LDM-RSIC at different values of T . The results indicate a significant improvement in the performance of LDM-RSIC as the step is increased to 3. Once T reaches a value of 4, the performance of LDM-RSIC stabilizes. Considering the trade-off between time complexity and performance, we set the

total diffusion step T to 4 in this paper. Moreover, our LDM-RSIC enjoys faster convergence speed compared to traditional DMs, which typically require more than 100 iterations. This accelerated convergence can be attributed to the deployment of the DM exclusively on low-dimensional latent spaces.

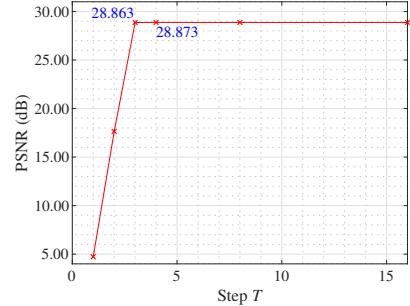


Fig. 7. Ablation study of the total diffusion step T of the proposed LDM-RSIC on the testing set of DOTA.

2) *Effectiveness of Prior Information*: To verify whether the prior information introduced by DFAM benefits the enhancement processing of LDM-RSIC, we conduct experiments on images “P0216” and “P0253” from the DOTA testing set under different values of λ . Table I presents the results of the proposed LDM-RSIC with and without DFAM. As shown in the results, with DFAM, the PSNR gains are approximately 0.2 dB higher than the baseline in all cases of λ , indicating that LDM indeed generates useful prior information to improve decoding performance.

TABLE I
DECODING PERFORMANCE OF THE PROPOSED LDM-RSIC WITH AND WITHOUT DFAM ON THE DOTA TESTING SET

λ	Metrics	Baseline	w/o DFAM	w/ DFAM
4×10^{-4}	PSNR \uparrow	27.17	27.25	27.37
	MS-SSIM \uparrow	0.8646	0.8663	0.8685
	LPIPS \downarrow	0.4135	0.4111	0.405
	PSNR \uparrow	28.65	28.78	28.87
8×10^{-4}	MS-SSIM \uparrow	0.9006	0.9024	0.9036
	LPIPS \downarrow	0.3661	0.3657	0.3594

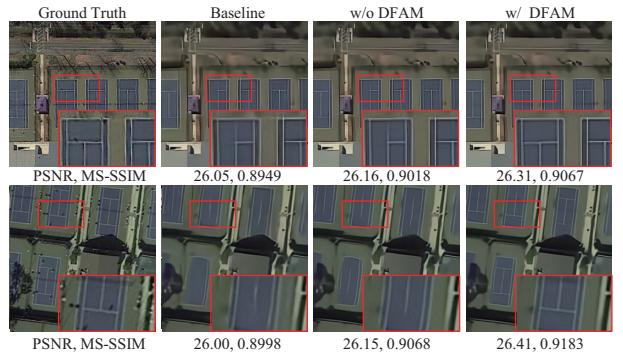


Fig. 8. Visualization of the decoded images “P0216” and “P0253” of the baseline and the proposed LDM-RSIC with (w/) and without (w/o) DFAM.

Furthermore, to investigate whether the LDM-based generated prior contributes to recovering additional image details,

Fig. 8 presents the decoded images with different model components under $\lambda = 8e - 4$. From the results, it is evident that without DFAM, the texture in the decoded image “P0216” appears more blurred as well as crippled. Although there is a 0.11 dB PSNR gain compared to the baseline, it fails to generate additional edge details. In contrast, with DFAM, the decoded images exhibit more impressive edge texture and better structural information. The image “P0253” further demonstrates these results.

In summary, we can demonstrate the effectiveness of the generated prior in providing more detailed information, thus enhancing the RD performance of RS image compression algorithms.

3) *Effectiveness of LDM-based Scheme*: In order to evaluate the effectiveness of the LDM-based scheme, we show the visualization of the decoded images obtained from LDM-RSIC and JPEG2000* obtained by employing the proposed LDM-based scheme to enhance the learning-based image compression algorithm ELIC and the traditional compression algorithm JPEG2000, respectively.

Enhancing ELIC: We compare the baseline with the proposed LDM-RSCI, and visual results are depicted in Fig. 9. Notably, LDM-RSCI yields finer texture details compared to the baseline across various bpp values. For instance, at low bitrates (*e.g.*, bpp = 0.0522), the baseline’s decoded vehicles appear blurry, whereas LDM-RSCI successfully restores the structure information. At higher bitrates (*e.g.*, bpp = 0.2615), LDM-RSCI not only enhances vehicle contour clarity but also recovers road markings.



Fig. 9. Visualization of the decoded images of the baseline (1st row) and the proposed LDM-RSCI (2nd row). PSNR and MS-SSIM are used for quality evaluation.

Enhancing JPEG2000: As depicted in Fig. 10, JPEG2000 decoded images exhibit noticeable artifacts. However, employing the LDM-based scheme significantly enhances texture detail clarity. For instance, at low bitrates (*e.g.*, bpp = 0.2000), JPEG2000 decoded image exhibits severe artifacts, whereas the enhanced version JPEG2000* enhances the image, resulting in a 1.17 dB increase in PSNR. This enhancement notably clarifies road markings. Similarly, at higher bitrates, the structure of the images recovered with JPEG2000* is significantly clearer.

In conclusion, the experiments demonstrate the positive impact of the LDM-based scheme on both deep learning-based

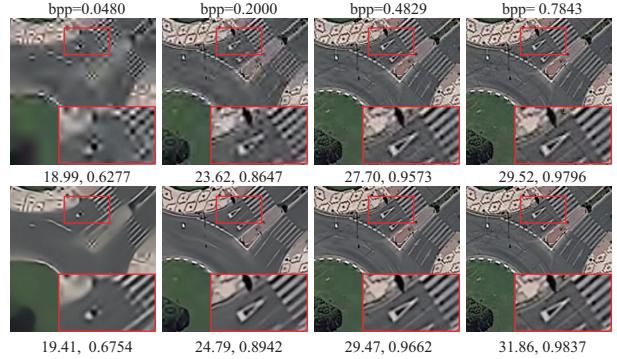


Fig. 10. Visualization of the decoded images of the JPEG2000 (1st row) and the proposed JPEG2000* (2nd row).

and traditional image compression algorithms for decoding high-quality images.

F. Model Complexity

The complexity of the proposed LDM-RSIC model is evaluated in comparison to other state-of-the-art image compression algorithms, as shown in Table II. The DOTA testing set is utilized for the experiment. The complexity is measured in terms of FLOPs (floating-point operations), the number of parameters, and encoding/decoding times on both CPU and GPU platforms.

In terms of FLOPs, LDM-RSIC requires 94.41G, placing it in the higher range compared to lighter models like Entroformer (44.76G) but lower than the computationally expensive WeConvne (150.76G). Similarly, the number of parameters for LDM-RSIC is 79.53 M, which is significantly higher than Entroformer (12.67M) but still lower than WeConvne (105.51M). This indicates that while LDM-RSIC has a relatively complex architecture, it is not the most computationally heavy among the compared models.

For encoding and decoding times, LDM-RSIC performs efficiently on the GPU, with encoding and decoding times of 0.4671s and 0.4413s, respectively. These results show that while LDM-RSIC has higher computational requirements than lightweight models such as STF and Entroformer, it is still capable of producing high-quality compressed images with reasonable computational costs. On the CPU, the encoding and decoding times are 1.7934s and 1.8530s, respectively, which are comparable to other models with similar complexities.

In conclusion, LDM-RSIC demonstrates moderate-to-high computational complexity, reflecting its ability to capture more intricate image features compared to simpler models, while maintaining reasonable inference times, especially on GPU platforms. This trade-off between complexity and performance is appropriate for applications requiring high-quality image compression without excessive computational overhead.

V. CONCLUSION

In this paper, we develop the LDM-RSIC to enhance the RD performance of the learning-based image compression algorithm ELIC. Specifically, LDM-RSIC utilizes the LDM to

TABLE II
COMPLEXITY COMPARISON OF SEVERAL IMAGE COMPRESSION ALGORITHMS.

Methods	FLOPs (G)	Parameters (M)	CPU		GPU	
			Encoding Time (s)	Decoding Time (s)	Encoding Time (s)	Decoding Time (s)
Entroformer [33]	44.76	12.67	1.3482	0.5018	0.2762	0.0919
STF [53]	99.83	33.35	1.1395	1.3175	0.1343	0.1879
ELIC [11]	31.66	54.46	1.7934	1.6796	0.4671	0.2507
LIC-TCM [36]	35.23	44.97	22.0959	21.0951	0.4637	0.4045
WeConvene [55]	150.76	105.51	50.6482	49.3341	0.3472	0.3880
LDM-RSIC (Ours)	94.41	79.53	1.7934	1.8530	0.4671	0.4413

generate compression distortion prior, which is then integrated into the Transformer-based MEN to enhance the quality of the decoded images. Additionally, we propose a channel attention and gate-based DFAM to better utilize the prior. Furthermore, we apply the proposed LDM-based scheme to enhance the traditional image compression algorithm JPEG2000, significantly improving the perceptual quality of the decoded images. Extensive experiments on two widely used RS image datasets demonstrate that LDM-RSIC significantly outperforms state-of-the-art traditional and learning-based image compression algorithms in terms of objective performance and subjective perception quality.

REFERENCES

- [1] P. Han, B. Zhao, and X. Li, “Edge-guided remote-sensing image compression,” *IEEE Trans. Geosci. Remote Sensing*, vol. 61, pp. 1–15, 2023.
- [2] T. Pan, L. Zhang, L. Qu, and Y. Liu, “A coupled compression generation network for remote-sensing images at extremely low bitrates,” *IEEE Trans. Geosci. Remote Sensing*, vol. 61, pp. 1–14, 2023.
- [3] J. Li and X. Hou, “Object-fidelity remote sensing image compression with content-weighted bitrate allocation and patch-based local attention,” *IEEE Trans. Geosci. Remote Sensing*, vol. 62, pp. 1–14, 2024.
- [4] D. S. Taubman, M. W. Marcellin, and M. Rabbani, “JPEG2000: Image compression fundamentals, standards and practice,” *J. Electron. Imaging*, vol. 11, no. 2, pp. 286–287, 2002.
- [5] F. Bellard, “BPG image format.” [Online]. Available: <https://bellard.org/bpg>
- [6] “Versatile video coding reference software version 16.0 (vtm-16.0).” [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM-/releases/VTM-16.0
- [7] T. Pan, L. Zhang, Y. Song, and Y. Liu, “Hybrid attention compression network with light graph attention module for remote sensing images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.
- [8] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learned image compression with discretized Gaussian mixture likelihoods and attention modules,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 7939–7948.
- [9] Q. Mao, C. Wang, M. Wang, S. Wang, R. Chen, L. Jin, and S. Ma, “Scalable face image coding via stylegan prior: Toward compression for human-machine collaborative vision,” *IEEE Trans. Image Process.*, vol. 33, pp. 408–422, 2024.
- [10] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, “Variational image compression with a scale hyperprior,” in *Int. Conf. on Learning Representations (ICLR)*, 2018.
- [11] 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 *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 5718–5727.
- [12] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, “End-to-End Learnt Image Compression via Non-Local Attention Optimization and Improved Context Modeling,” *IEEE Trans. Image Process.*, vol. 30, pp. 3179–3191, Jan. 2021.
- [13] H. Fu, F. Liang, J. Lin, B. Li, M. Akbari, J. Liang, G. Zhang, D. Liu, C. Tu, and J. Han, “Learned image compression with gaussian-laplacian-logistic mixture model and concatenated residual modules,” *IEEE Trans. Image Process.*, vol. 32, pp. 2063–2076, 2023.
- [14] S. Xiang and Q. Liang, “Remote sensing image compression with long-range convolution and improved non-local attention model,” *Signal Process.*, vol. 209, p. 109005, 2023.
- [15] X. Lu, B. Wang, X. Zheng, and X. Li, “Exploring models and data for remote sensing image caption generation,” *IEEE Trans. Geosci. Remote Sensing*, vol. 56, no. 4, pp. 2183–2195, 2017.
- [16] L. Zhang, X. Hu, T. Pan, and L. Zhang, “Global priors with anchored-stripe attention and multiscale convolution for remote sensing image compression,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 138–149, 2024.
- [17] H. Wang, L. Liao, J. Xiao, W. Lin, and M. Wang, “Uplink-assist downlink remote-sensing image compression via historical referencing,” *IEEE Trans. Geosci. Remote Sensing*, vol. 61, pp. 1–15, 2023.
- [18] S. Xiang and Q. Liang, “Remote sensing image compression based on high-frequency and low-frequency components,” *IEEE Trans. Geosci. Remote Sensing*, vol. 62, pp. 1–15, 2024.
- [19] X. Dong, Z. Mao, Y. Sun, and X. Xu, “Short-term wind power scenario generation based on conditional latent diffusion models,” *IEEE Trans. Sustain. Energy*, vol. 15, no. 2, pp. 1074–1085, 2024.
- [20] R. Wu, T. Yang, L. Sun, Z. Zhang, S. Li, and L. Zhang, “Seesr: Towards semantics-aware real-world image super-resolution,” *arXiv preprint arXiv:2311.16518*, 2023.
- [21] Z. Chen, Y. Zhang, D. Liu, J. Gu, L. Kong, X. Yuan *et al.*, “Hierarchical integration diffusion model for realistic image deblurring,” *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2024.
- [22] C. Corneanu, R. Gadde, and A. M. Martinez, “Latentpaint: Image inpainting in latent space with diffusion models,” in *IEEE Winter Conf. on Appl. of Comput. Vis. (WACV)*, January 2024, pp. 4334–4343.
- [23] Y. Wang, H. Liu, Y. Feng, Z. Li, X. Wu, and C. Zhu, “Headdiff: Exploring rotation uncertainty with diffusion models for head pose estimation,” *IEEE Trans. Image Process.*, vol. 33, pp. 1868–1882, 2024.
- [24] C. Wang, Z. Hao, Y. Tang, J. Guo, Y. Yang, K. Han, and Y. Wang, “Sam-diffssr: Structure-modulated diffusion model for image super-resolution,” *arXiv preprint arXiv:2402.17133*, 2024.
- [25] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, “Repaint: Inpainting using denoising diffusion probabilistic models,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 11 461–11 471.
- [26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 10 684–10 695.
- [27] Y. Qian, Z. Tan, X. Sun, M. Lin, D. Li, Z. Sun, L. Hao, and R. Jin, “Learning accurate entropy model with global reference for image compression,” in *Int. Conf. Learn. Represent. (ICLR)*, 2020.
- [28] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” in *Int. Conf. Learn. Represent. (ICLR)*, 2017.
- [29] Y. Hu, W. Yang, Z. Ma, and J. Liu, “Learning end-to-end lossy image compression: A benchmark,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4194–4211, 2021.
- [30] 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, 2022.
- [31] Y. Xie, K. L. Cheng, and Q. Chen, “Enhanced invertible encoding for learned image compression,” in *ACM Int'l Conf. on Multimedia (ACM MM)*, 2021, pp. 162–170.

- [32] G. Gao, P. You, R. Pan, S. Han, Y. Zhang, Y. Dai, and H. Lee, “Neural image compression via attentional multi-scale back projection and frequency decomposition,” in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 14 677–14 686.
- [33] Y. Qian, X. Sun, M. Lin, Z. Tan, and R. Jin, “Entroformer: A transformer-based entropy model for learned image compression,” in *Int. Conf. Learn. Represent. (ICLR)*, 2022.
- [34] J. Lee, S. Cho, and S.-K. Beack, “Context-adaptive entropy model for end-to-end optimized image compression,” in *Int. Conf. Learn. Represent. (ICLR)*, 2018.
- [35] M. Li, K. Ma, J. You, D. Zhang, and W. Zuo, “Efficient and effective context-based convolutional entropy modeling for image compression,” *IEEE Trans. Image Process.*, vol. 29, pp. 5900–5911, 2020.
- [36] J. Liu, H. Sun, and J. Katto, “Learned image compression with mixed Transformer-CNN architectures,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 14 388–14 397.
- [37] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, “Diffusion models in vision: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10 850–10 869, 2023.
- [38] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong, “Uni-controlnet: All-in-one control to text-to-image diffusion models,” *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2024.
- [39] M. Ren, M. Delbracio, H. Talebi, G. Gerig, and P. Milanfar, “Multiscale structure guided diffusion for image deblurring,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 10 721–10 733.
- [40] A. Lin, B. Chen, J. Xu, Z. Zhang, G. Lu, and D. Zhang, “DS-TransUNet: Dual Swin Transformer U-Net for medical image segmentation,” *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022.
- [41] B. Xia, Y. Zhang, S. Wang, Y. Wang, X. Wu, Y. Tian, W. Yang, and L. Van Gool, “Diffir: Efficient diffusion model for image restoration,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 13 049–13 059.
- [42] K. Shah, S. Chen, and A. Klivans, “Learning mixtures of Gaussians using the DDPM objective,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023, pp. 19 636–19 649.
- [43] Y. Wen, X. Ma, X. Zhang, and M.-O. Pun, “GCD-DDPM: A generative change detection model based on difference-feature-guided DDPM,” *IEEE Trans. Geosci. Remote Sensing*, vol. 62, pp. 1–16, 2024.
- [44] H. Gu, X. Zhang, J. Li, H. Wei, B. Li, and X. Huang, “Federated learning vulnerabilities: Privacy attacks with denoising diffusion probabilistic models,” in *Proceedings of the ACM Web Conf.*, New York, NY, USA, 2024, p. 1149–1157.
- [45] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in *Int. Conf. Learn. Represent. (ICLR)*, 2014.
- [46] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.
- [47] J. Ding, N. Xue, Y. Long, G. Xia, and Q. Lu, “Learning RoI Transformer for detecting oriented objects in aerial images,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2019.
- [48] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, 2010, pp. 270–279.
- [49] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [50] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” in *Int. Conf. Learn. Represent. (ICLR)*, 2017.
- [51] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 586–595.
- [52] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” *ITU-T VCEG-M33*, Apr., 2001.
- [53] R. Zou, C. Song, and Z. Zhang, “The devil is in the details: Window-based attention for image compression,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 17 492–17 501.
- [54] R. Yang and S. Mandt, “Lossy image compression with conditional diffusion models,” in *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 64 971–64 995.
- [55] H. Fu, J. Liang, Z. Fang, J. Han, F. Liang, and G. Zhang, “Weconvene: Learned image compression with wavelet-domain convolution and entropy model,” *Proc. European Conf. on Computer Vision (ECCV)*, 2024.