

UNIFYING GENERATION AND COMPRESSION: ULTRA-LOW BITRATE IMAGE CODING VIA MULTI-STAGE TRANSFORMER

Naifu Xue¹, Qi Mao^{1*}, Zijian Wang¹, Yuan Zhang¹, Siwei Ma²

¹Communication University of China, ²Peking University

ABSTRACT

Recent progress in generative compression technology has significantly improved the perceptual quality of compressed data. However, these advancements primarily focus on producing high-frequency details, often overlooking the ability of generative models to capture the prior distribution of image content, thus impeding further bitrate reduction in extreme compression scenarios (< 0.05 bpp). Motivated by the capabilities of predictive language models for lossless compression, this paper introduces a novel *Unified Image Generation-Compression* (UIGC) paradigm, merging the processes of generation and compression. A key feature of the UIGC framework is the adoption of vector-quantized (VQ) image models for tokenization, alongside a multi-stage transformer designed to exploit spatial contextual information for modeling the prior distribution. As such, the dual-purpose framework effectively utilizes the learned prior for entropy estimation and assists in the regeneration of lost tokens. Extensive experiments demonstrate the superiority of the proposed UIGC framework over existing codecs in perceptual quality and human perception, particularly in ultra-low bitrate scenarios (≤ 0.03 bpp), pioneering a new direction in generative compression.

Index Terms— Generative Compression, Extreme Compression, Image Generation, VQGANs, Transformer

1. INTRODUCTION

Ultra-low bitrate compression presents a significant challenge in the field of image/video compression, particularly due to substantial information loss when faced with extremely limited network bandwidth, such as in satellite communications. Traditional block-based compression codecs, *e.g.*, VVC [1], are constrained to use large quantization steps in such scenarios, inevitably leading to noticeable blurring and blocking artifacts. Despite the superior rate-distortion (R-D) performance of learning-based compression techniques [2–5], these methods produce blurry images at low bitrates, due to the reliance on optimization of pixel-oriented distortion metrics.

To address challenges in ultra-low bitrate scenarios, generative compression methods [6–14] have employed gener-

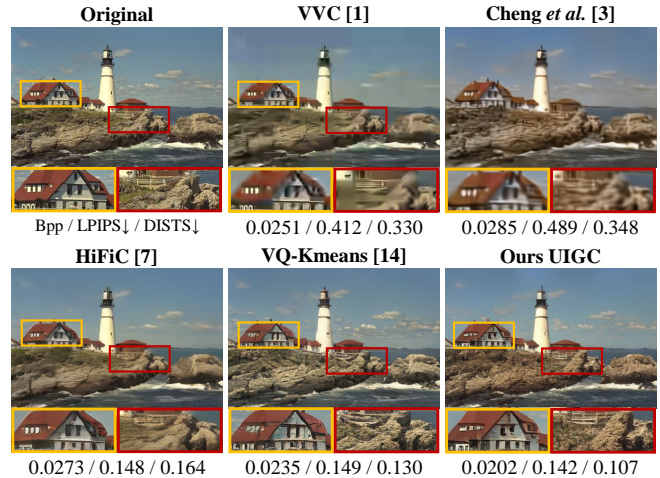


Fig. 1: Qualitative comparisons between state-of-the-art image compression methods, including traditional [1], learning-based [3], generative-based [7, 14], and Ours.

ative models [15–17] to enhance the visual quality of decoded images, with a focus primarily on the generator’s ability to *produce high-frequency details*. This paradigm follows two primary technical pathways: one involves training existing end-to-end image codecs using perceptual and adversarial losses [6–9], and the other [10–14] leverages specially designed encoders to compress images into more compact representations. However, despite their effectiveness, these methods tend to overlook *modeling the prior distribution of image content*, a critical aspect that differentiates image generation from image reconstruction task. In situations where significant information loss occurs due to extremely limited bandwidth, it is plausible to reconstruct some of the lost content by sampling from the prior distribution.

Meanwhile, the fundamental aspect of entropy estimation requires accurately determining the prior probability distribution of symbols, thereby boosting the efficiency of entropy coding. Consequently, the mathematical equivalence in estimating this *prior probability distribution*—between entropy minimization in lossless compression and \log_2 -likelihood maximization in generation—poses a critical inquiry: *Is it possible to develop a method that models the prior distribu-*

*THE CORRESPONDING AUTHOR.

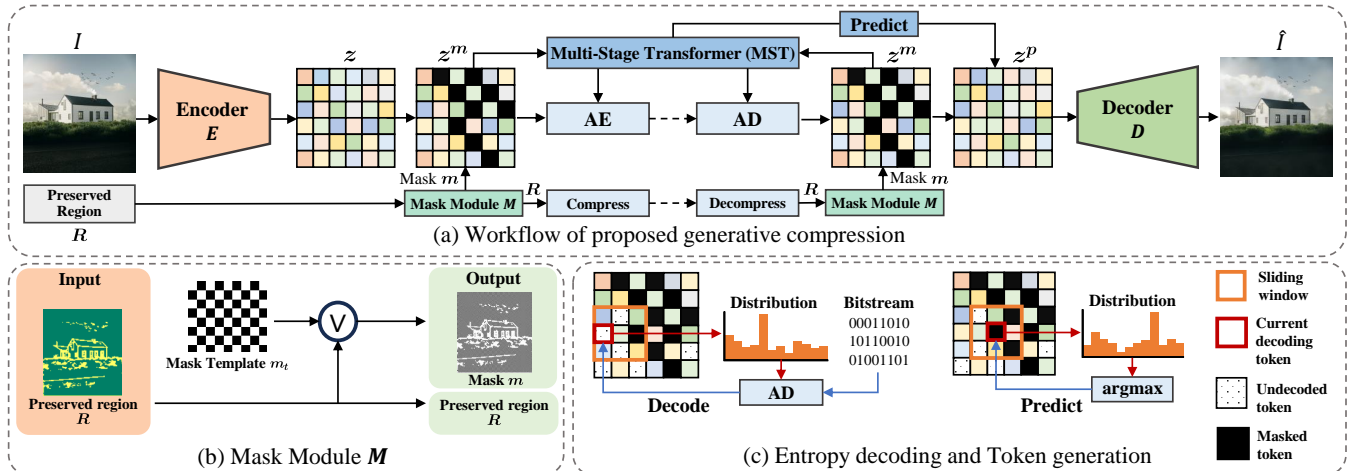


Fig. 2: Overview of the proposed UIGC framework. (a) the overall compression workflow: we adopt a multi-stage transformer, and AE/AD denote arithmetic encoder/decoder; (b) the mask mechanism using mask module M , and \vee denotes the logical “OR” operator; (c) entropy decoding and token generation on the decoder side.

tion for both entropy estimation in compression and sampling in generation, all within a cohesive and unified framework?

Recently, in the field of natural language process (NLP), Delétang *et al.* [18] have demonstrated that sequence generation models, such as large language models (LLMs), can be effectively used for lossless compression. Nevertheless, the extensive representation space of images presents a significant challenge in efficiently modeling the prior distribution. On the brighter side, advancements in Vector-Quantized Image Modeling (VIM) [19, 20] have made strides in compressing images into compact and discrete token representations using the vector-quantized (VQ) encoder. This development paves the way for transforming images into compact and discrete token representations via VIM, which enables the utilization of discrete generative models, similar to LLMs, for both entropy estimation and token generation.

In this work, we present a novel image compression paradigm, the Unified Image Generation-Compression (UIGC) framework, innovatively designed to facilitate *both entropy encoding of tokens and the prediction of lost tokens*. By converting images into discrete token representations using VIM [19], our UIGC codec focuses on accurate prior modeling of these tokens and the strategic discarding of nonessential tokens, leading to enhanced bitrate reduction while still producing perceptually pleasing images. Departing from the traditional autoregressive [19] and non-autoregressive [20] models typically used in NLP, we propose a **Multi-Stage Transformer (MST)** specifically tailored to image characteristics. The MST restructures the autoregressive order by dividing the token map into four groups, enabling most tokens to effectively utilize the surrounding context for prediction. Recognizing the crucial role of structural information in visual perception, we incorporate an edge-preserved checkerboard mask pattern, which selectively dis-

cards tokens while maintaining essential structural details. With MST’s multi-stage order, the prediction of lost tokens is significantly enhanced, utilizing the surrounding content to ensure the generation of high-quality images.

To evaluate the efficiency of the proposed UIGC framework, we conduct experiments on the Kodak [21] and CLIC [22] datasets. The experimental results, both quantitative and qualitative, demonstrate that our method surpasses existing techniques in maintaining perceptual quality under ultra-low bitrate conditions (≤ 0.03 bpp). As shown in Fig. 1, our framework effectively reduces the bitrate while maintaining uncompromising image quality.

2. UNIFYING GENERATION AND COMPRESSION

In this work, we aim to compress the image I at ultra-low bitrates while reconstructing the image \hat{I} with a pleasing perceptual quality. In contrast to previous generative compression approaches that predominantly concentrate on the reconstruction of high-frequency details, our proposed UIGC framework shifts its emphasis toward modeling the prior distribution of image content for *both entropy estimation and content generation*. Fig. 2 presents the overview of the proposed method. In the following sections, we provide a detailed explanation of the image coding methodologies in Section 2.1. Subsequently, we introduce the proposed Multi-Stage Transformer (MST) in Section 2.2 and the edge-preserved checkerboard mask in Section 2.3 to leverage spatial contextual dependencies in image token maps and effectively eliminate redundant tokens. This approach facilitates efficient prior modeling and bitrate savings while preserving image quality.

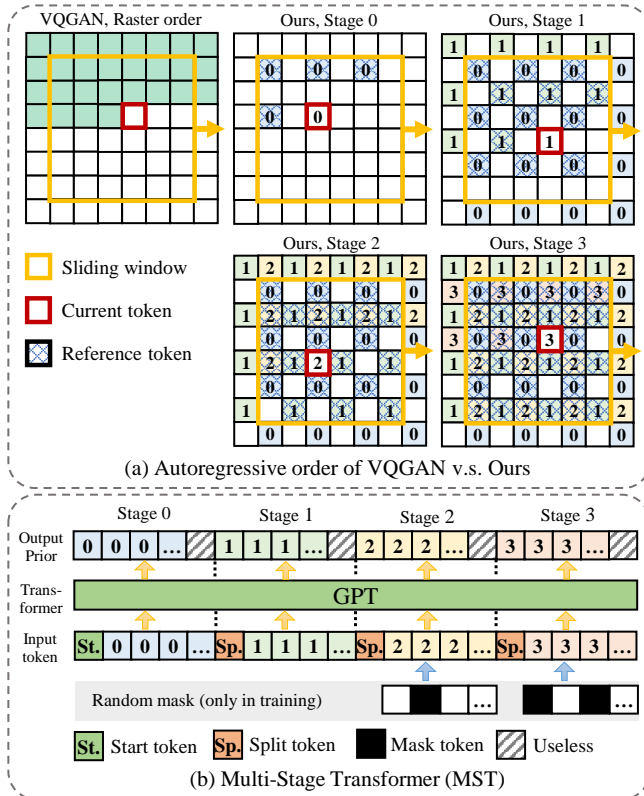


Fig. 3: Our MST. We implement a GPT-style transformer for the prior modeling at each stage.

2.1. Image Coding via UIGC

The primary goal of our UIGC framework is to leverage the prior distribution for both entropy estimation and content generation. To achieve this, we selectively compress essential tokens, discarding redundant ones to achieve bitrate savings. The discarded tokens are then generated directly on the decoder side. As illustrated in Fig. 2(a), on the encoder side, the VQ encoder E [19] transforms the given image I into a token representation $z \in \mathbb{N}^{h \times w}$. Then, we discard redundant tokens using the mask mechanism in Section 2.3, where the mask $m \in \{0, 1\}^{h \times w}$ replaces discarded tokens as [Mask]:

$$z_{i,j}^m = m_{i,j} z_{i,j} + (1 - m_{i,j})[\text{Mask}]. \quad i \in h, j \in w \quad (1)$$

Subsequently, the masked token map z^m is compressed by the arithmetic compression codec using the prior distribution of the MST, while the [Mask] tokens are skipped. Since the mask m is generated by our Mask Module M according to the preserved region R , we further losslessly compress this region and transmit it to the decoder side.

On the decoder side, we restore the layout of the token map z^m using the decoded mask m . As illustrated in Fig. 2(c), the unmasked tokens undergo decompression using the prior distribution from the MST. Simultaneously, the

masked tokens are predicted with the highest probability from this prior distribution. Thus, the z^p is derived as:

$$z_{i,j}^p = m_{i,j} z_{i,j}^m + (1 - m_{i,j}) (\arg\max p_{i,j}). \quad (2)$$

Finally, the VQ decoder D [19] utilizes z^p to reconstruct the decoded image \hat{I} .

2.2. Multi-Stage Transformer

As illustrated in Fig. 3(a), the transformer in VQGAN [19] employs a sliding window for raster order autoregressive encoding to manage memory usage. However, this design limits the current encoding position to consider only its *upper-left* context for prediction, potentially compromising the accuracy of prior modeling. Recognizing the spatial correlation dependencies in images, we introduce the MST inspired by the multi-stage grouping algorithm in [5] to enhance this accuracy by rearranging the autoregressive order. In particular, the token map is partitioned into four groups, and each group undergoes processing in raster order using a sliding window, as presented in Fig. 3(a).

Consequently, the MST is structured into four distinct stages. In Stage 0, tokens in Group 0 are sequentially encoded, with each token referencing only its upper-left content within the group. Following this, Stage 1 encodes tokens in Group 1, allowing each token to reference surrounding tokens in Group 0 and upper-left tokens in Group 1. This sequential encoding pattern persists in Stages 2 and 3, enabling each token to reference surrounding tokens in preceding groups and upper-left tokens in the current group. Each group of tokens in the sliding window is flattened in raster order and inputted into each stage of the transformer, as depicted in Fig. 3(b). We utilize a GPT-style transformer for autoregressive encoding, aiming to maximize the \log_2 -likelihood of tokens defined as:

$$L_{\text{transformer}} = \mathbb{E}_{I \sim p(I)} \left[\sum_{k=0}^{h \times w} -\log_2 p(z_k | z_{\leq k}^m) \right]. \quad (3)$$

During training, a random mask is applied to groups 2-3 to simulate lost tokens, and the transformer estimates the categorical distribution as the prior.

2.3. Edge-Preserved Checkerboard Mask

Another essential concern is the strategic discarding of redundant tokens to achieve bitrate savings without compromising image quality. In our proposed MST, tokens in Group 0 and Group 1 serve as anchors, providing surrounding references for all tokens in the subsequent groups. Hence, a *checkerboard pattern* is used as the mask template, retaining all tokens in both groups to ensure accurate prior modeling. Furthermore, tokens associated with the object structure are preserved, recognizing their crucial role in visual perception. As such, in this section, we propose an edge-preserving

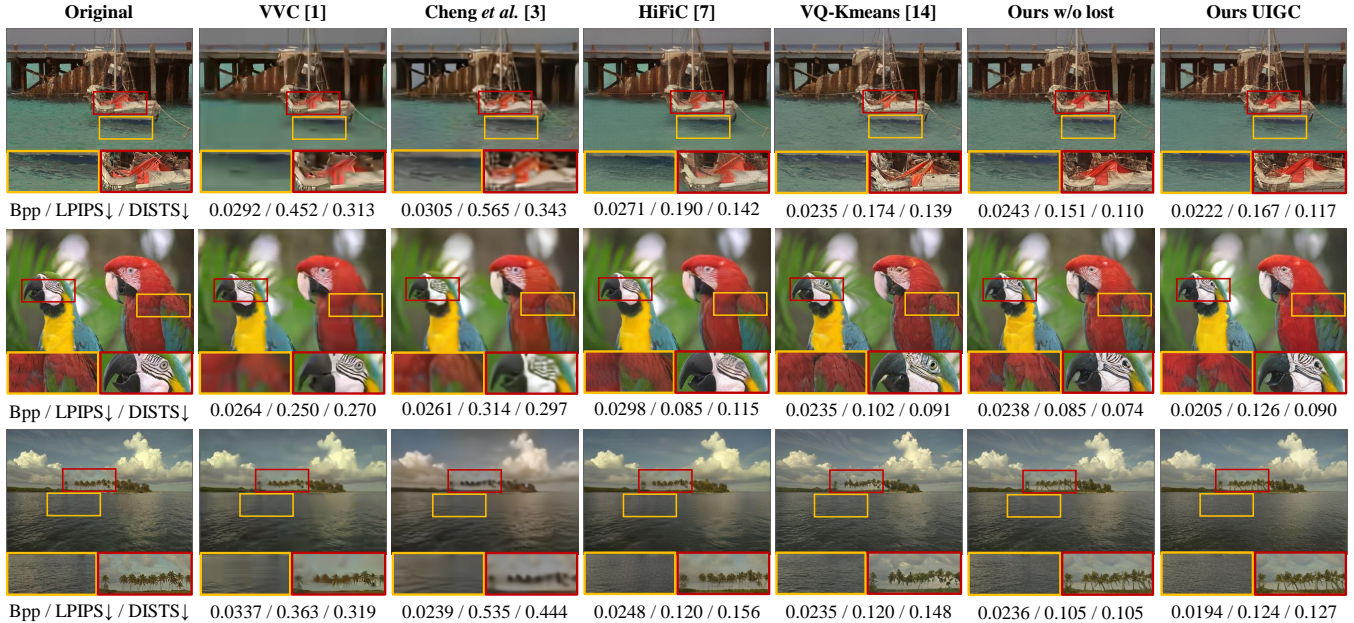


Fig. 4: Qualitative comparisons on the Kodak dataset [21]. In particular, ↓ indicates that lower is better.

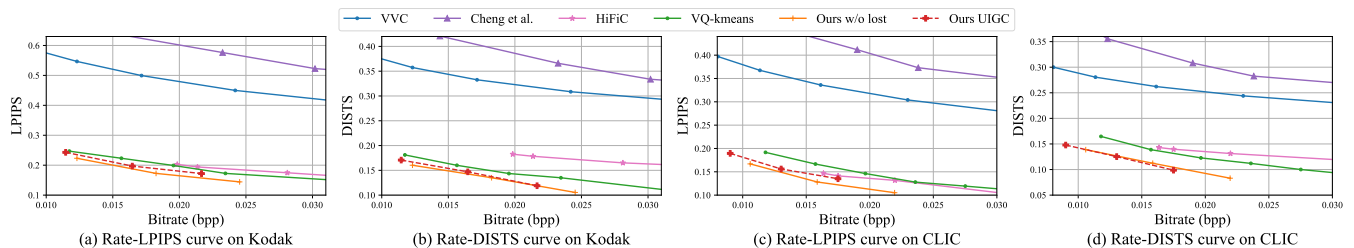


Fig. 5: R-D curves on the Kodak [21] and the CLIC [22] datasets. Ours w/o lost: ours without token lost and generation.

checkerboard mask mechanism as demonstrated in Fig. 2(b). First, we extract the object structure using the edge extractor [23], regarding it as the preserved region $R \in \{0, 1\}^{h \times w}$. Then, our proposed mask module M generates the final mask $m = m_t \vee R$, where the $m_t \in \{0, 1\}^{h \times w}$ represents the checkerboard template. The positions of Group 0 and Group 1 are set to 1 in the template m_t to retain the corresponding tokens. Our supplementary material provides further details.

3. EXPERIMENTS

3.1. Implementation Details

We employ the architecture of encoder and decoder from VQGAN [19], and utilize the K-means clustering method detailed in [14] to fine-tune the officially provided pre-trained model with a codebook size of 16384, yielding models with codebook sizes of $\{16, 64, 256\}$ (denoted as VQ16, VQ64, and VQ256) suitable for ultra-low bitrates. During the VQ-codec training, we utilize the default settings and training losses as in [19]. For the MST, we set the size of the sliding

Table 1: Average BD-LPIPS↓/DISTS↓ gains on the Kodak [21] and the CLIC [22] datasets. Anchor: VVC [1].

Method	Kodak		CLIC	
	LPIPS	DISTS	LPIPS	DISTS
Cheng et al. [3]	0.041	0.044	0.043	0.054
HiFiC [7]	-0.260	-0.134	-0.176	-0.113
VQ-kmeans [14]	-0.288	-0.182	-0.170	-0.127
Ours w/o lost	-0.321	-0.199	-0.207	-0.151
Ours UIGC	-0.310	-0.195	-0.198	-0.150

window at 18×18 . We train the proposed model on the ImageNet dataset [24]. To evaluate the performance of the proposed model, we use two widely recognized datasets in image compression: the Kodak [21] and the CLIC datasets [22].

3.2. Compression Performance Evaluation

Compared Methods. To assess the effectiveness of our proposed framework, we conduct a benchmark against both traditional standard and neural-based compression frameworks,

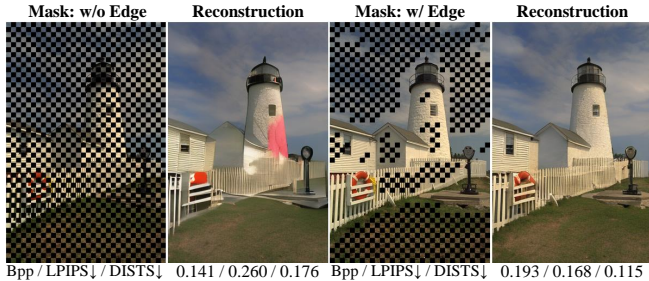


Fig. 6: Visual comparisons between checkerboard mask with and without edge preservation.

Table 2: R-D performance of RT and MST on the Kodak dataset [21]. The codec is VQ16 and VQ256.

Method	Bpp	LPIPS↓	DISTS↓
VQ16 RT	0.0113	0.2540	0.1712
VQ16 MST	0.0115	0.2429	0.1707
VQ256 RT	0.0216	0.1895	0.1226
VQ256 MST	0.0217	0.1720	0.1189

including the latest VVC [1] codec, the typical end-to-end learning-based approach by Cheng *et al.* [3] (MS-SSIM optimized, for better perceptual quality), and generative image compression codecs, such as HiFiC [7], as well as the VQGAN-based codec VQ-Kmeans [14]. Furthermore, we develop an additional baseline denoted as “Ours w/o lost”, which directly applies the MST for entropy estimation without lost tokens and prediction.

Quantitative Evaluation. Rather than relying on traditional objective quality assessments like PSNR and SSIM, we incorporate recent perceptual quality-based metrics, such as Learned Perceptual Image Patch Similarity (LPIPS) and Deep Image Structure and Texture Similarity (DISTS), as they offer closer alignment with human perception of images. Additionally, we employ bits per pixel (bpp) as a metric to evaluate the rate performance. We present the R-D performance in Fig. 5. We also evaluate the R-D performance improvement using VVC as an anchor with Bjontegaard-Delta metric [25]. In particular, we adopt BD-LPIPS and BD-DISTS metrics in Table 1, which represent the average perceptual quality improvement under the equivalent bitrate. It can be clearly observed that our proposed UIGC exhibits superior R-D performance, delivering enhanced visual quality in ultra-low bitrate scenarios ($\text{bpp} \leq 0.03$). Note that VQ-kmeans [14] utilizes the same VQ codec as ours. However, its R-D performance is inferior due to the absence of entropy estimation. This finding underscores the effectiveness of the UIGC method, which integrates both entropy estimation and content generation through the use of the prior distribution. While there is a slight performance drop compared to “Ours w/o lost” due to the generated content being slightly different from the real image, the UIGC framework further reduces bitrate and achieves

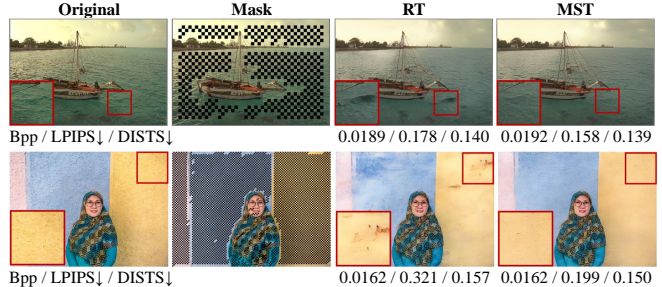


Fig. 7: Visual Comparisons between MST and RT.

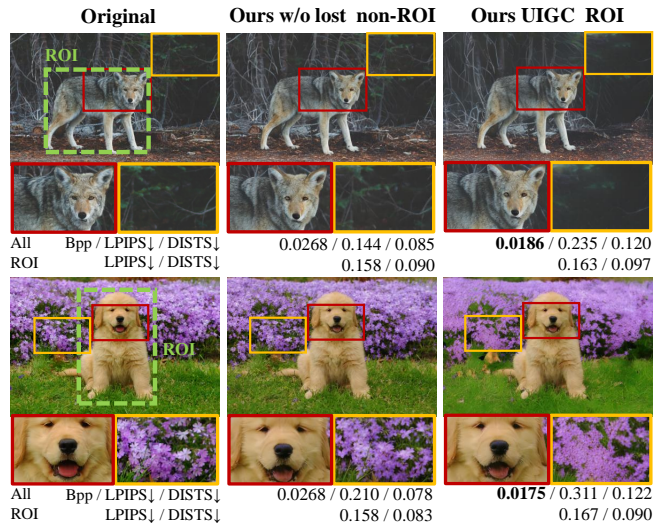


Fig. 8: Examples of ROI coding. “Ours UIGC ROI” ensures ROI quality while significantly lowering bitrate.

comparable perceptual quality as shown in Fig. 4.

Qualitative Evaluation. Fig. 4 shows the reconstruction results of various methods on the Kodak dataset [21], with the corresponding bpp, LPIPS, and DISTS. At ultra-low bitrates, VVC [1] and Cheng *et al.* [3] exhibit severe blurring. Although HiFiC [7] and VQ-Kmeans [14] have improved image quality, the detail is not satisfactory: HiFiC shows the grid artifact on the sea and the abnormal reconstruction of the parrot’s eye; VQ-Kmeans produces distorted structure on the boat. In contrast, our methods exhibit more natural sea surface and object structures (*e.g.*, the boat, parrots, trees). Moreover, the UIGC further saves bitrate while maintaining almost the same visual quality as “Ours w/o lost”, with only a negligible loss in water texture details.

3.3. Ablation Studies and Discussion

Efficiency of the MST. We conduct experiments to ascertain the advancements of the proposed MST over the Raster Transformer (RT) detailed in [19]. Table 2 indicates that while RT and MST exhibit similar levels of entropy coding efficiency, the MST surpasses RT in the perceptual quality of the gener-

ated images. Fig. 7 presents examples of images generated by both MST and RT. The RT tends to create unnatural textures in areas like the sea and walls due to its predominant reliance on upper-left positional references, resulting in a lower visual quality compared to MST.

Mask Pattern. To evaluate the importance of tokens associated with the object structure, we test two mask patterns: checkerboard with and without edge preservation. Fig. 6 demonstrates that although excluding edge tokens contributes to further bitrate reduction, it simultaneously causes issues like distorted edges and abnormal content (for instance, the red region on the tower). This observation effectively confirms the essentiality of implementing an edge preservation mechanism in our approach.

Region of Interest Compression. Region of interest (ROI) coding, essential in multimedia applications, requires high-quality compression of selected regions while allowing for more aggressive compression in non-essential areas to reduce bitrate. Our proposed UIGC framework is adept at accommodating this need by selectively preserving tokens in the ROI. As shown in Fig. 8, UIGC’s approach to ROI coding not only significantly reduces the bitrate but also maintains an aesthetically pleasing visual quality in the regions of interest.

4. CONCLUSIONS

In this work, we propose a novel UIGC paradigm, specifically tailored for ultra-low bitrate image compression. This versatile framework adeptly utilizes the learned prior distribution for both entropy estimation and the regeneration of lost tokens. We further design the MST to boost prior modeling accuracy, and introduce an edge-preserving checkerboard mask pattern to discard unnecessary tokens for bitrate saving. Our experimental results validate the UIGC’s superiority over existing codecs in visual quality, particularly in ultra-low bitrate (≤ 0.03 bpp) scenarios. We believe that the UIGC scheme represents a significant advancement in generative compression, charting a new course for future developments.

5. REFERENCES

- [1] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm, “Overview of the versatile video coding (vvc) standard and its applications,” *TCSVT*, 2021.
- [2] David Minnen, Johannes Ballé, and George D Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” in *NIPS*, 2018.
- [3] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto, “Learned image compression with discretized gaussian mixture likelihoods and attention modules,” in *CVPR*, June 2020.
- [4] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin, “Checkerboard context model for efficient learned image compression,” in *CVPR*, 2021.
- [5] Ming Lu, Fangdong Chen, Shiliang Pu, and Zhan Ma, “High-efficiency lossy image coding through adaptive neighborhood information aggregation,” *arXiv preprint*, 2022.
- [6] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool, “Generative adversarial networks for extreme learned image compression,” in *ICCV*, 2019.
- [7] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson, “High-fidelity generative image compression,” in *NIPS*, 2020.
- [8] Shoma Iwai, Tomo Miyazaki, Yoshihiro Sugaya, and Shinichiro Omachi, “Fidelity-controllable extreme image compression with generative adversarial networks,” in *ICPR*, 2021.
- [9] Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer, “Multi-realism image compression with a conditional generator,” in *CVPR*, 2023.
- [10] Jianhui Chang, Qi Mao, Zhenghui Zhao, Shanshe Wang, Shiqi Wang, Hong Zhu, and Siwei Ma, “Layered conceptual image compression via deep semantic synthesis,” in *ICIP*, 2019.
- [11] Jianhui Chang, Zhenghui Zhao, Chuanmin Jia, Shiqi Wang, Lingbo Yang, Qi Mao, Jian Zhang, and Siwei Ma, “Conceptual compression via deep structure and texture synthesis,” *TIP*, 2022.
- [12] Jianhui Chang, Zhenghui Zhao, Lingbo Yang, Chuanmin Jia, Jian Zhang, and Siwei Ma, “Thousand to one: Semantic prior modeling for conceptual coding,” in *ICME*, 2021.
- [13] Jianhui Chang, Jian Zhang, Jiguo Li, Shiqi Wang, Qi Mao, Chuanmin Jia, Siwei Ma, and Wen Gao, “Semantic-aware visual decomposition for image coding,” *IJCV*, 2023.
- [14] Qi Mao, Tinghan Yang, Yinuo Zhang, Zijian Wang, Meng Wang, Shiqi Wang, and Siwei Ma, “Extreme image compression using fine-tuned vqgans,” *arXiv preprint*, 2023.
- [15] Diederik P Kingma and Max Welling, “Auto-encoding variational bayes,” *arXiv preprint*, 2013.
- [16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” *NIPS*, vol. 27, 2014.
- [17] Jonathan Ho, Ajay Jain, and Pieter Abbeel, “Denoising diffusion probabilistic models,” *NIPS*, 2020.
- [18] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al., “Language modeling is compression,” *arXiv preprint*, 2023.
- [19] Patrick Esser, Robin Rombach, and Bjorn Ommer, “Taming transformers for high-resolution image synthesis,” in *CVPR*, 2021.
- [20] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman, “Maskgit: Masked generative image transformer,” in *CVPR*, 2022.
- [21] Eastman Kodak, “Kodak photocd dataset,” 2013.
- [22] Toderici George, Shi Wenzhe, Timofte Radu, Theis Lucas, Balle Johannes, Agustsson Eirikur, Johnston Nick, and Mentzer Fabian, “Workshop and challenge on learned image compression (clic2020),” 2020.
- [23] Saining Xie and Zhuowen Tu, “Holistically-nested edge detection,” in *ICCV*, 2015.

- [24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.
- [25] Gisle Bjontegaard, "Calculation of average psnr differences between rd-curves," *ITU SG16 Doc. VCEG-M33*, 2001.
- [26] Jean-loup Gailly and Mark Adler, "Zlib compression library," 2004.
- [27] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja, "Compressai: a pytorch library and evaluation platform for end-to-end compression research," *arXiv preprint arXiv:2011.03029*, 2020.

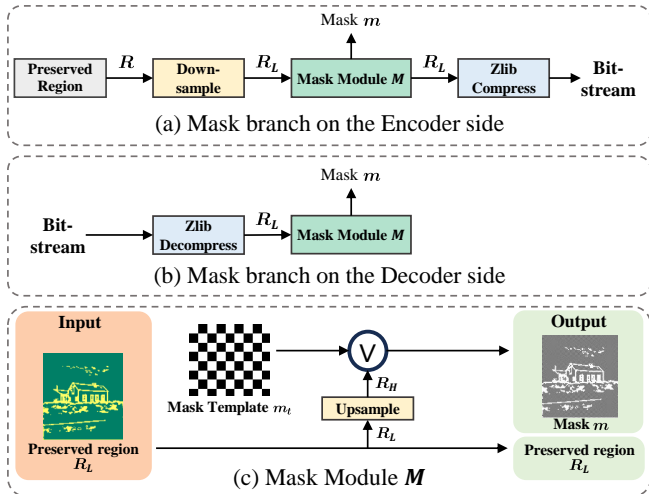


Fig. 9: Details of the downsampling and upsampling design in our mask mechanism. (a) the downsampling and compression of the preserved region R in the encoder side, as well as the mask generation; (b) the decompression of the R_L and mask generation in the decoder side; (c) upsampling design inside the Mask Module M .

A Implementation Details

A.1 Multi-Stage Transformer

The GPT model used in our Multi-Stage Transformer (MST) consists of 32 layers of transformers. Each layer of transformer has 16 heads of self-attention and the embedding dimension is 1152.

During training, image patches of size 288 are randomly cropped. These patches are encoded into an 18×18 token map by VQGAN’s Encoder [19], matching MST’s sliding window size. Additionally, masks are randomly applied to the tokens of Group 2 and Group 3 with a variable probability ranging from 0% to 100%.

A.2 Edge-Preserved Checkerboard Mask

In our design, we downsample the preserved region $R \in \{0, 1\}^{h \times w}$ into $R_L \in \{0, 1\}^{0.5h \times 0.5w}$ to reduce bit cost. The encoder-side downsampling and compression of R is depicted in Fig. 9(a), while Fig. 9(b) demonstrates the decoder-side decompression of R_L . Given that R_L only contains 0 and 1, we consider it as a bitstream and apply lossless compression using the Zlib codec [26]. Our Mask Module M then upsamples the R_L into $R_H \in \{0, 1\}^{h \times w}$ and generates the final mask m , as illustrated in Fig. 9 (c).

B Experimental Details

B.1 Traditional Codec

We use the VTM version 22.2 as the implementation of the latest traditional compression standard VVC [1]. The following commands are used for the VVC codec:

```
EncoderApp
```

```
-i {input file name}
-c encoder_intra_vtm.cfg
-q {QP}
-o /dev/null
-b {bitstream file name}
-wdt {image width}
-hgt {image height}
-fr 1
-f 1
--InputChromaFormat=444
--InputBitDepth=8
--ConformanceWindowMode=1
DecoderApp
-b {bitstream file name}
-o {reconstruction file name}
-d 8
```

B.2 Neural-based Codec

All neural-based codecs are trained on the ImageNet dataset [24]. The implementation details are as follows:

- **Cheng et al. [3]:** We employ the CompressAI library [27] implementation, adhering to its default losses and training strategies. The hyperparameter λ is adjusted for ultra-low bitrate compression training.
- **HiFiC [7]:** We utilize the pytorch implementation of the HiFiC [7], available at [Justin-Tan/high-fidelity-generative-compression](#). The default losses and training strategies are maintained for model training, with the hyperparameter λ adjusted for ultra-low bitrate compression.
- **VQ-kmeans [14]:** This method is implemented based on the pre-trained VQGAN [19] model with a codebook size of 16384, generating models with codebook sizes of {8, 16, 32, 64, 128, 256}. The training settings and losses follow the method described in [14].

C. Visual Results

We provide CLIC reconstructions for qualitative comparison, as Fig. 10 shows.

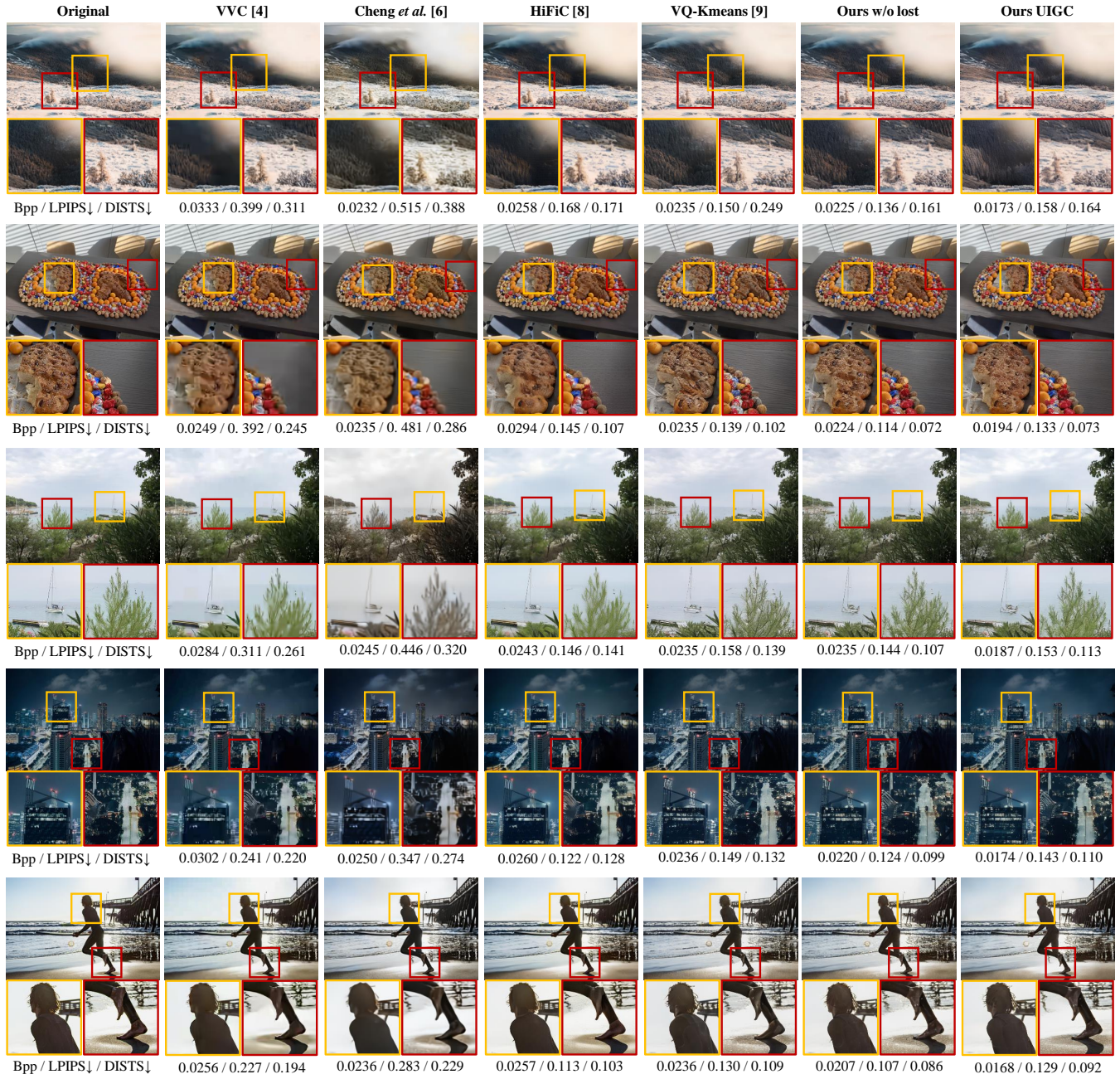


Fig. 10: Qualitative comparisons on the CLIC dataset [22]. In particular, ↓ indicates that lower is better.