Decouple Distortion from Perception: Region Adaptive Diffusion for Extreme-low Bitrate Perception Image Compression

Supplementary Material

In this document, we provide supplementary material for the proposed MRIDC, including framework implementation details, more experimental content and results, a discussion of the limitations of existing schemes, and an introduction to application scenarios. Our code can be found at https://github.com/xjc97/mridc.

1. Implementation Details

The proposed MRIDC framework achieves DP balance for image compression at extremely low bitrates by leveraging vector quantization latent and diffusion model generation. Utilizing the introduced RoI prior, MRIDC dynamically adjusts coding resource allocation within each image, effectively balancing the minimization of local distortion in the RoI with the maximization of overall perceptual quality. This section details the design and implementation of the Encoder and Decoder components of MRIDC.

1.1. Encoder

The encoder comprises two modules, f_l and f_h, designed to extract information from the original image. Both modules are based on a VQGAN architecture, with the specific structure detailed in Table 1. The compression rate is adjusted by tuning the number of downsampling steps m.

We employed a pre-trained VQGAN encoder for f_l and developed a smaller, self-trained encoder for f_h, utilizing the vector-quantize-pytorch library. The f_l module comprises 72.142M parameters, trained exclusively on the ImageNet dataset, with a codebook containing 1024 codes. It reduces a 256×256 (resp. 512×512) image to a 16×16 (resp. 32×32) token representation. In contrast, the f_h module, containing 4M parameters, is trained on the OpenImage-v6 dataset, employs a codebook with 256 codes, and compresses a 64×64 latent representation into 8×8 tokens.

1.2. Decoder

The decoder incorporates three key modules: MPT, CM, and a diffusion model. The MPT is fine-tuned from a pre-trained bidirectional transformer on the OpenImage-v6 dataset. The bidirectional transformer within MPT has 174.161M parameters on ImageNet $256 \times 256$and 176.307M parameters on ImageNet 512×512. The hyperparameter settings are detailed in Table 2. The pre-trained model is optimized for token unmasking using a cross-entropy loss with label smoothing set to 0.1. AdamW is employed as the optimizer, with a learning rate of $1e^{-4}$, betas = $(0.9, 0.96)$, and a weight decay of $1e^{-5}$. An arccos scheduler is applied for masking during training, irrespective of image resolution. Furthermore, 10% of the conditions are dropped for classifier-free guidance.

The CM module is based on the ControlNet model. It uses $\hat{z}_l$ as the conditional input and is trained from scratch on the OpenImage-v6 dataset. The diffusion decoder, built on a pre-trained Stable Diffusion v2.1 model, remains frozen during the training process.

Table 1. High-level architecture of the encoder of our VQGAN. Note that $h = \frac{H}{2^m}$, $w = \frac{W}{2^m}$ and $f = 2^m$

Architecture	Number	Size
input x	1	$\mathbb{R}^{H \times W \times C}$
Conv2D	1	$\mathbb{R}^{H \times W \times C'}$
{Rsidual Block, Downsample Block}	m	$\mathbb{R}^{h \times w \times C''}$
Residual Block	1	$\mathbb{R}^{h \times w \times C''}$
Non-Local Block	1	$\mathbb{R}^{h \times w \times C''}$
Residual Block	1	$\mathbb{R}^{h \times w \times C''}$
{GroupNorm, Swish, Conv2D}	1	$\mathbb{R}^{h \times w \times n_z}$

Table 2. Bidirectional transformer architecture in MPT.

Parameter	Setting
Hidden Dimension	768
Codebook Size	1024
Depth	24
Attention Heads	16
MLP Size	3072
Dropout	0.1

1.3. Bitstream Analysis

Since the encoding end uses vector quantization, the size of the compressed latent feature can be predefined and fixed. On this basis, arithmetic coding is utilized to calculate the corresponding bitstream size R:

$$R = -\alpha \log_2(\alpha) + (1-\alpha) \log_2(\frac{1-\alpha}{K}), \qquad (1)$$

https://github.com/CompVis/taming-transformers

https://github.com/lucidrains/vector-quantize-pytorch

https://github.com/valeoai/Maskgit-pytorch

https://github.com/huggingface/diffusers

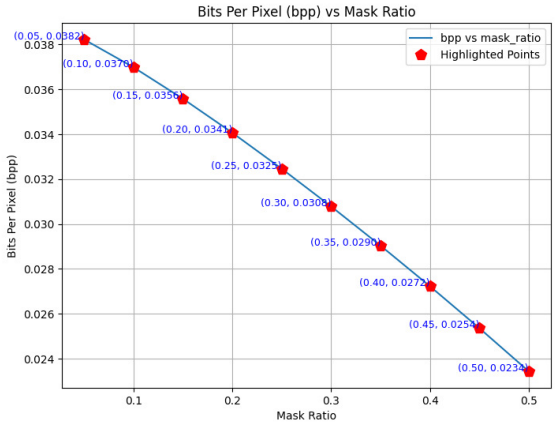

Figure 1. Relationship between bpp and mask ratio.

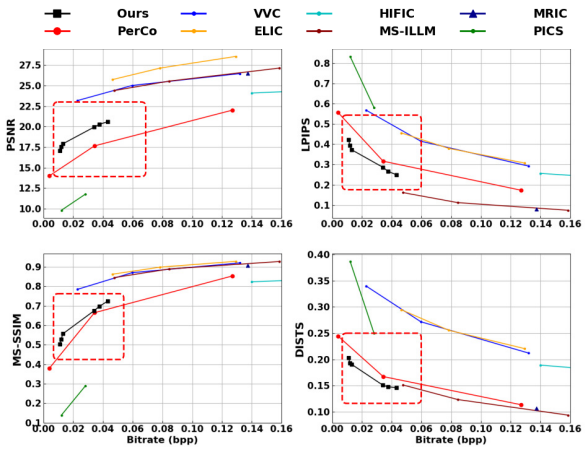

Figure 2. Quality results on Kodak dataset.

where α represents the mask ratio, K denotes the codebook size, and the final bits per pixel (bpp) is calculated by:

$$bpp = \frac{\alpha R}{W \times H}, \tag{2}$$

where W and H are the width and height of the original image. The bitstream contains two parts: $\bar{z}_l \in \mathbb{R}^{H_{z_l} \times W_{z_l} \times 256}$ and $z_h \in \mathbb{R}^{8\times8\times320}$. Based on Equation (1) and Equation (2), the bitstream size of z_h is a fixed value 0.0019 bpp and the bitstream size of z_l is controlled by downsample factor m and mask ratio α.

In our implementation, we set $W' \times H' = 512 \times 512$, $m = 32$ to yield $z_l^{lo} \in \mathbb{R}^{16\times16\times256}$, corresponding to a bitrate of 0.01 bpp, while setting $s = 16$ produces $z_l^{hi} \in \mathbb{R}^{32\times32\times256}$, which corresponds to a higher bitrate of 0.04 bpp. Additionally, various bitrates can be achieved by adjusting the mask ratio α without retraining the model. The relationship between the mask ratio and the bpp of z_l is depicted in Figure 1. As the mask ratio increases, the bpp decreases accordingly.

2. Experiments

2.1. RoI-based Metric Calculation

We use the clean-fid library to compute FID and KID. LPIPS is calculated using the original github. DISTS is calculated using github. MS-SSIM is calculated using the FAIR Neural Compression library. To evaluate and compare the performance metrics of various mask types and mask ratios, we randomly selected 411 images from the MSCOCO30K dataset, where the bbox size ranges from 30% to 40% of the entire image. The RoI is defined as the area within the bbox, and distortion metrics are calculated specifically for this region to assess the quality of the RoI. For fair testing and comparison, the dataset will be made publicly available.

https://github.com/GaParmar/clean-fid

https://github.com/facebookresearch/NeuralCompression

https://github.com/dingkeyan93/DISTS

https://github.com/facebookresearch/NeuralCompression

Table 3. Quality results of text guidance. MS stands for MS-SSIM.

	bpp	PSNR	MS	LPIPS	DISTS
w/o text	0.0397	20.8	0.73	0.25	0.15
w/ text	0.0427	20.5	0.72	0.26	0.15

2.2. Quantitative Results

Figure 2 presents the compression quality comparison results on the Kodak dataset. Due to the limited number of samples in the Kodak dataset (24 images), it is not feasible to compute perceptual quality metrics (FID and KID). Therefore, the comparison is restricted to objective and subjective distortion metrics. The results demonstrate that MRIDC outperforms both PerCo and PICS across all distortion metrics, indicating superior compression quality.

2.3. Impact of Text Guidance

To incorporate additional global semantic information, we experiment with an image-conditioned model, BLIP-2, to generate image captions for conditioning the diffusion decoder. The maximum caption length is constrained to 32 tokens, and the text captions are encoded using arithmetic coding.

Our findings indicate that text guidance enhances perceptual quality but also results in a slight increase in local image

https://huggingface.co/docs/transformers/main/model_doc/blip-2

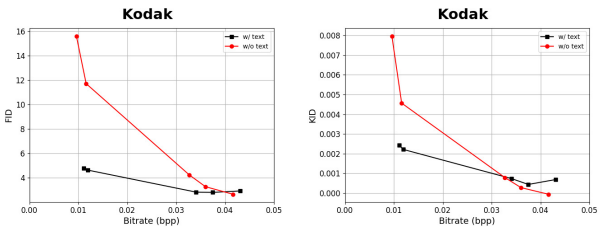

Figure 3. Impact of text guidance.

distortion. As shown in Figure 3, text guidance improves FID and KID metrics when the bitrate is below 0.03 bpp, with the effect being more pronounced at lower bitrates. Table 3 presents both objective and subjective distortion metrics, demonstrating that while text guidance improves perceptual quality, it causes a marginal increase in distortion. From the perspective of achieving DP balance and optimizing bitrate usage, text conditions were not incorporated in our main implementation.

2.4. Visual Results

We provide additional decoded images to compare the compression performance of MRIDC against state-of-the-art methods. Figure 4 highlights the comparison in terms of overall perceptual quality and local detail consistency. Figures 5 and 6 showcase decoded images from the MSCOCO30K dataset corresponding to MRIDC with small size z_l^{lo} and large size z_l^{hi}, respectively. Similarly, Figures 7 and 8 present decoded images from the Kodak dataset for MRIDC with small size z_l^{lo} and large size z_l^{hi}.

3. Discussion

While the proposed MRIDC demonstrates excellent DP quality balance at extreme-low bitrates, there remains potential for improvement in bitrate scalability. Currently, the implemented bitrate intervals are primarily determined by z_l, with its two main sizes corresponding to the compression performance of MRIDC at two specific bitrate points. Variable bitrate operations are achieved through the latent mask around these baseline bitrate points. Future work will focus on integrating the implementation and optimization of z_l into the end-to-end training of the overall framework, enabling support for a broader bitrate range and facilitating more precise bitrate adjustments.

4. Applications

The proposed MRIDC framework is specifically designed to address extreme-low bitrate compression scenarios, such as those encountered in satellite communications and remote sensing applications. In such cases, the images to be transmitted must be compressed to an exceptionally low bitrate due to the severe bandwidth constraints of ground-to-air communication channels. Despite these limitations, it is crucial to preserve key information within the images with minimal distortion, particularly in RoI specified by the user. This dual requirement of ultra-efficient compression and selective quality retention highlights the importance of MRIDC in ensuring both effective bandwidth utilization and the accurate representation of critical image content. Moreover, extending MRIDC to multi-modal data, including video and sensor information, could broaden its applicability, enabling it to serve as a comprehensive solution for future communication and data processing systems.

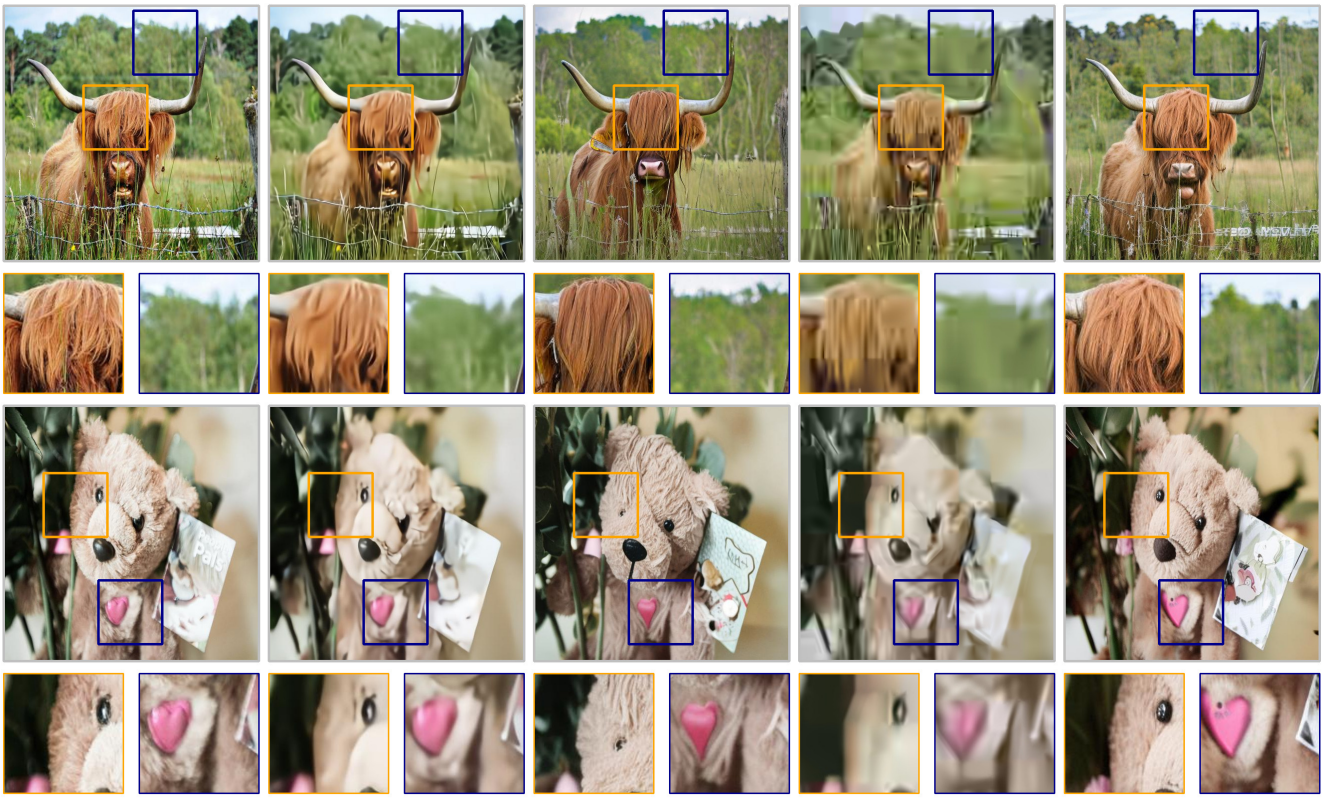

Figure 4. Visual comparison of decoded images with the state-of-the-art methods.

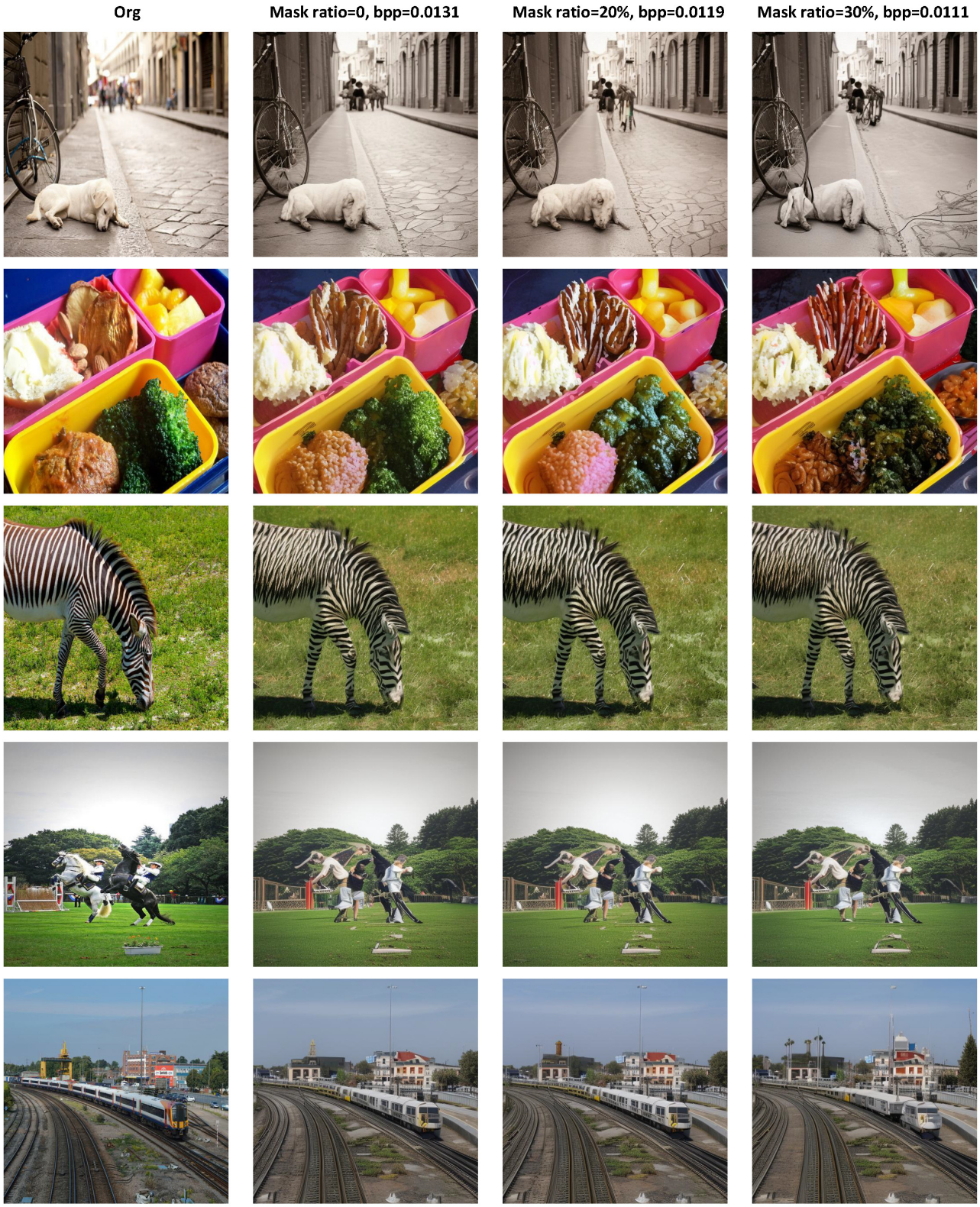

Figure 5. MRIDC compression results of MSCOCO30k samples on z_l^{lo}. The proportion of latent mask increases from left to right.

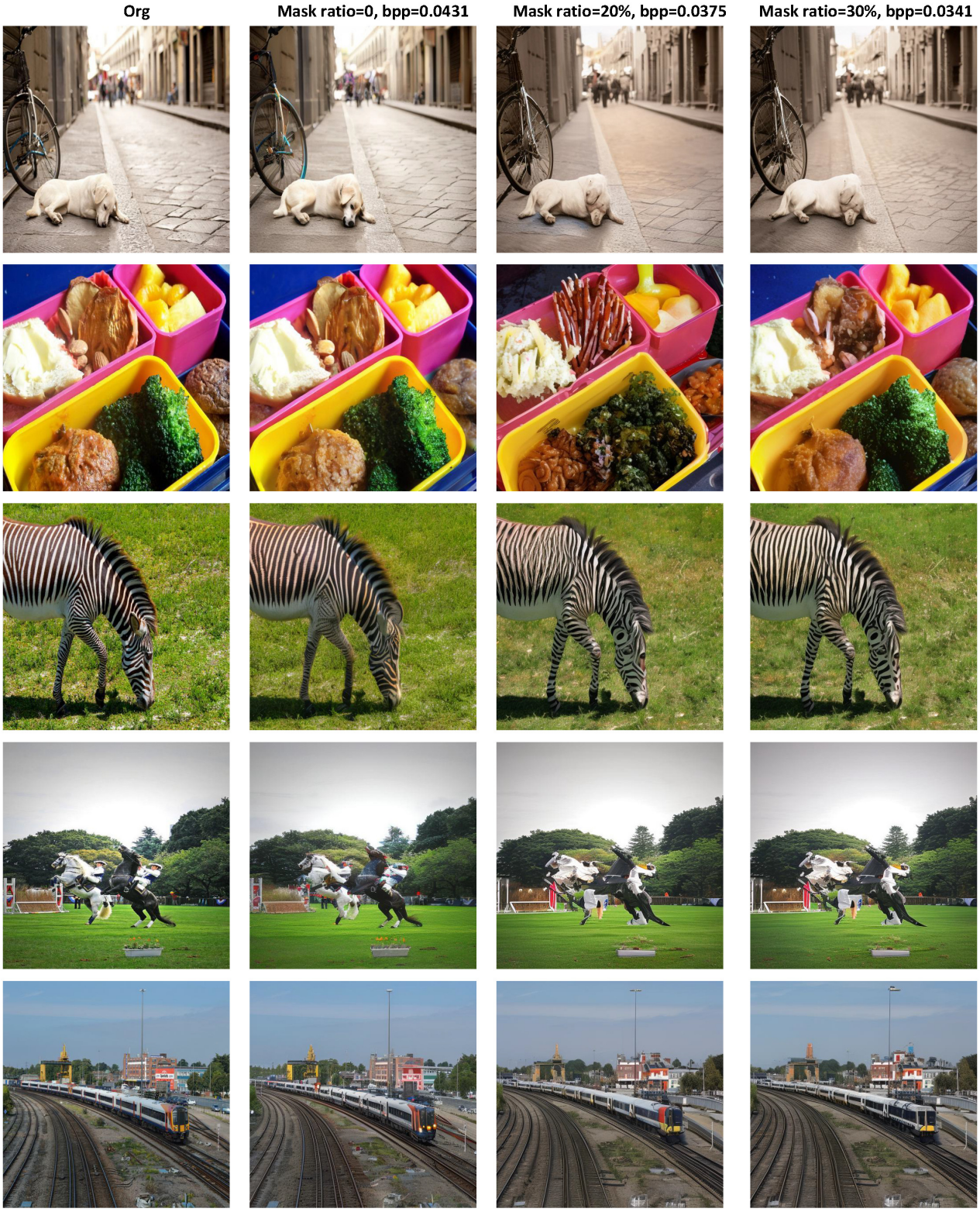

Figure 6. MRIDC compression results of MSCOCO30k samples on z_l^{hi}. The proportion of latent mask increases from left to right.

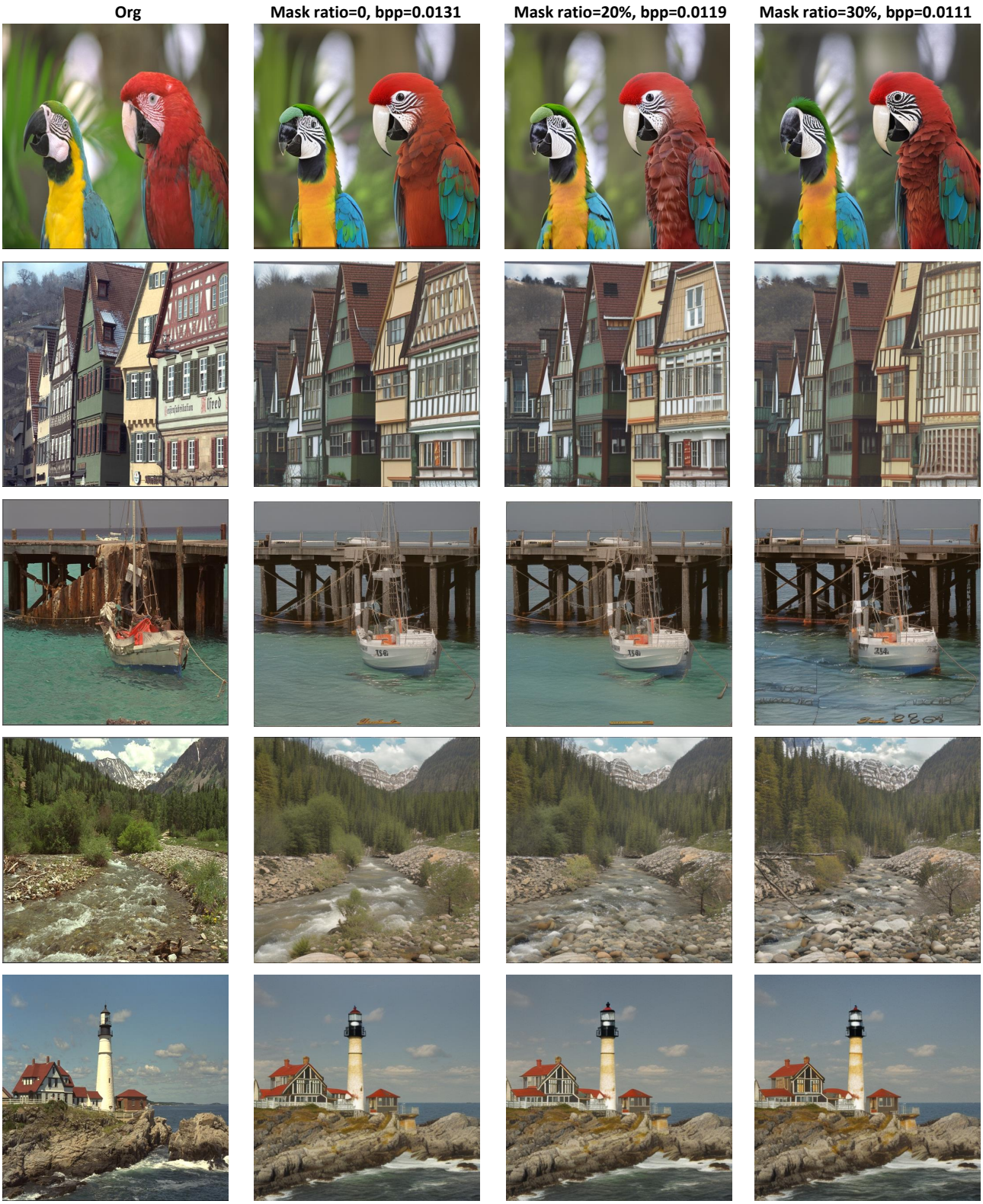

Figure 7. MRIDC compression results of Kodak samples on z_l^{lo}. The proportion of latent mask increases from left to right.

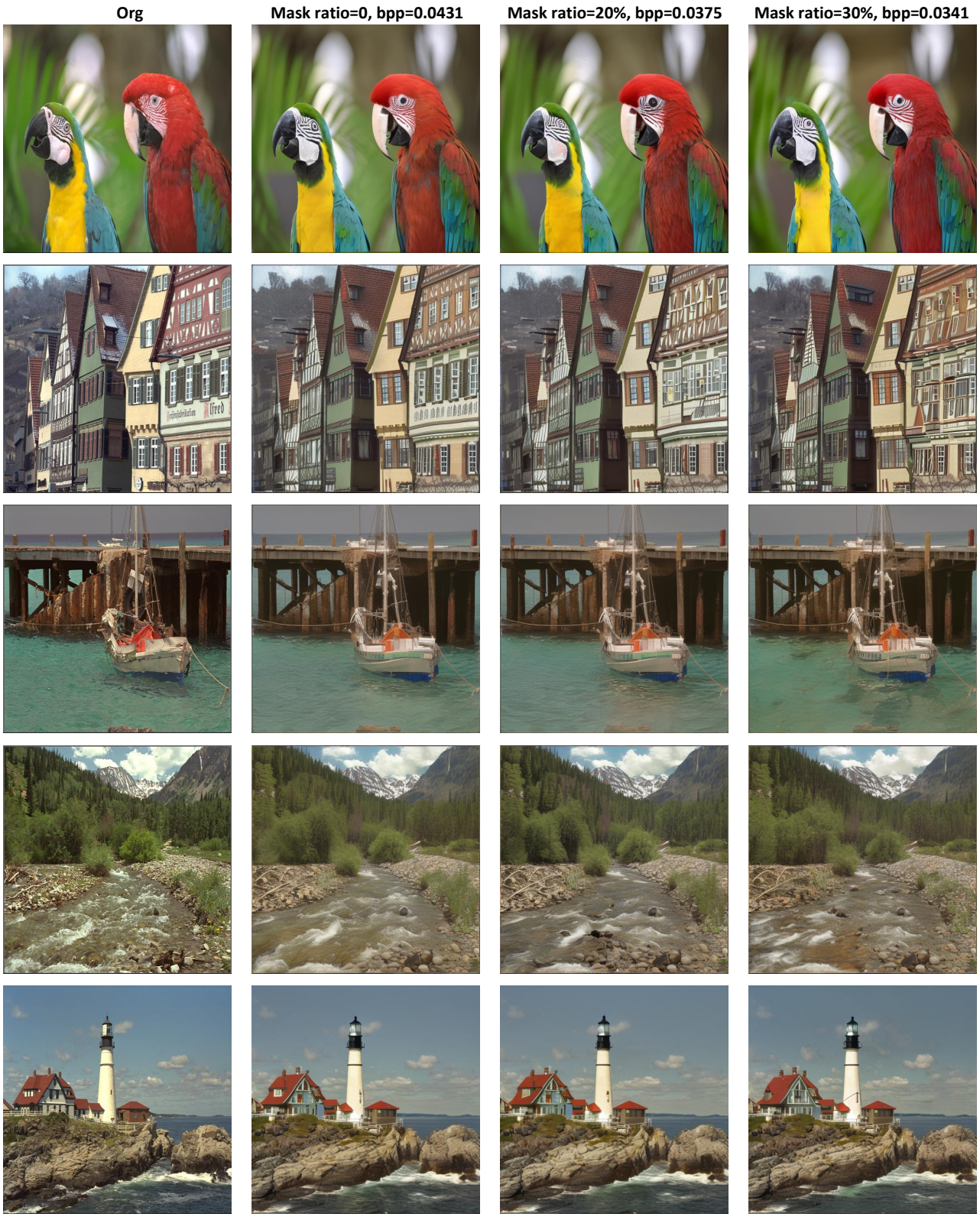

Figure 8. MRIDC compression results of Kodak samples on z_l^{hi}. The proportion of latent mask increases from left to right.