

A MATHEMATICAL DETAILS

Derivation of Eq. (4). First, according to Eq. (2), z_{n-1} can be sampled as:

$$z_{n-1} = \sqrt{\bar{\alpha}_{n-1}}(z_0 + \eta_{n-1}e) + \sqrt{1 - \bar{\alpha}_{n-1}}\epsilon_{n-1}, \quad (12)$$

$$= \sqrt{\bar{\alpha}_{n-1}}z_0 + \sqrt{\bar{\alpha}_{n-1}}\eta_{n-1}e + \underbrace{\sqrt{1 - \bar{\alpha}_{n-1}}\epsilon_{n-1}}_{\sim \mathcal{N}(0, (1 - \bar{\alpha}_{n-1})I)}, \quad (13)$$

where $\epsilon_{n-1} \sim \mathcal{N}(0, I)$. Second, for z_n defined in Eq. (2) and z_{n-1} defined in Eq. (3), we have:

$$z_{n-1} = k_n z_0 + m_n z_n + \sigma_n \epsilon, \quad (14)$$

$$= k_n z_0 + m_n (\sqrt{\bar{\alpha}_n}(z_0 + \eta_n e) + \sqrt{1 - \bar{\alpha}_n}\epsilon_n) + \sigma_n \epsilon, \quad (15)$$

$$= (k_n + m_n \sqrt{\bar{\alpha}_n})z_0 + m_n \sqrt{\bar{\alpha}_n} \eta_n e + \underbrace{m_n \sqrt{1 - \bar{\alpha}_n} \epsilon_n + \sigma_n \epsilon}_{\sim \mathcal{N}(0, (m_n^2(1 - \bar{\alpha}_n) + \sigma_n^2)I)}, \quad (16)$$

where $\epsilon_n \sim \mathcal{N}(0, I)$ and $\epsilon \sim \mathcal{N}(0, I)$. By combining Eq. (13) and Eq. (16), we obtain the following equations:

$$\begin{cases} \sqrt{\bar{\alpha}_{n-1}} = k_n + m_n \sqrt{\bar{\alpha}_n}, \\ \sqrt{\bar{\alpha}_{n-1}} \eta_{n-1} = m_n \sqrt{\bar{\alpha}_n} \eta_n, \\ 1 - \bar{\alpha}_{n-1} = m_n^2(1 - \bar{\alpha}_n) + \sigma_n^2. \end{cases} \quad (17)$$

Note that, referring to DDIM (Song et al., 2021), we set $\sigma_n = 0$ for simplicity. By solving Eq. (17), we have:

$$k_n = \sqrt{\bar{\alpha}_{n-1}} - \sqrt{\frac{1 - \bar{\alpha}_{n-1}}{1 - \bar{\alpha}_n}} \sqrt{\bar{\alpha}_n}, \quad m_n = \sqrt{\frac{1 - \bar{\alpha}_{n-1}}{1 - \bar{\alpha}_n}}, \quad \frac{\eta_n}{\eta_{n-1}} = \frac{\sqrt{1 - \bar{\alpha}_n} / \sqrt{\bar{\alpha}_n}}{\sqrt{1 - \bar{\alpha}_{n-1}} / \sqrt{\bar{\alpha}_{n-1}}}. \quad (18)$$

Therefore, η_n can be defined as:

$$\eta_n = \lambda \frac{\sqrt{1 - \bar{\alpha}_n}}{\sqrt{\bar{\alpha}_n}}, \quad (19)$$

where we set $\lambda = \frac{\sqrt{\bar{\alpha}_N}}{\sqrt{1 - \bar{\alpha}_N}}$ to ensure $\eta_N = 1$.

Derivation of Eq. (8). Substituting Eq. (6) into Eq. (7), we have:

$$\|z_0 - \hat{z}_0\|_2^2 = \left\| \left(\frac{z_n}{\sqrt{\bar{\alpha}_n}} - \frac{\sqrt{1 - \bar{\alpha}_n}}{\sqrt{\bar{\alpha}_n}} \tilde{\epsilon}_n \right) - \left(\frac{z_n}{\sqrt{\bar{\alpha}_n}} - \frac{\sqrt{1 - \bar{\alpha}_n}}{\sqrt{\bar{\alpha}_n}} \epsilon_\theta(z_n, c, n) \right) \right\|_2^2, \quad (20)$$

$$= \left\| \frac{\sqrt{1 - \bar{\alpha}_n}}{\sqrt{\bar{\alpha}_n}} \tilde{\epsilon}_n - \frac{\sqrt{1 - \bar{\alpha}_n}}{\sqrt{\bar{\alpha}_n}} \epsilon_\theta(z_n, c, n) \right\|_2^2, \quad (21)$$

$$= \frac{1 - \bar{\alpha}_n}{\bar{\alpha}_n} \|\tilde{\epsilon}_n - \epsilon_\theta(z_n, c, n)\|_2^2. \quad (22)$$

B EXPERIMENTAL DETAILS

Evaluation of third-party models. The quality factor of BPG (Bellard, 2014) was selected from {43, 45, 46, 48, 49, 51}. For VVC (Bross et al., 2021), we used the reference software VTM-23.0² with intra configuration. The quality factor was selected from the set {41, 43, 45, 47, 49, 52}. To compare ELIC (He et al., 2022) and HiFiC (Mentzer et al., 2020) at extremely low bitrates, we utilized their PyTorch implementation^{3,4} and retrained the model to achieve higher compression ratios, enabling a more direct comparison with our proposed method. For PerCo (Careil et al., 2024), since the official source codes and models are not available, we used a reproduced version⁵ as a substitute, which employs stable diffusion as the latent diffusion model. For MS-ILLM (Muckley et al., 2023), VQIR (Wei et al., 2024), Text+Sketch (Lei et al., 2023) and DiffEIC (Li et al., 2024b), we used the

²https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-23.0

³<https://github.com/JiangWeibeta/ELIC>

⁴<https://github.com/Justin-Tan/high-fidelity-generative-compression>

⁵<https://github.com/Nikolai10/PerCo/tree/master>

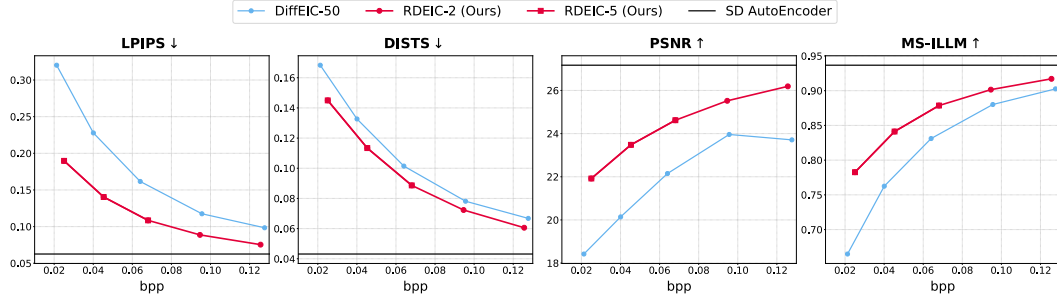


Figure 9: Quantitative performance on the MS-COCO 30k dataset.

publicly released checkpoints from their GitHub repositories, and used them for evaluation with the provided code.

Additional implementation details. We use Stable Diffusion 2.1-base as the specific implementation of stable diffusion. Throughout all our experiments, the weights of stable diffusion remain frozen. Similar to DiffEIC (Li et al., 2024b), the control module in our RDEIC has the same encoder and middle block architecture as stable diffusion and reduces the channel number to 20% of the original. The variance sequence $\{\beta_t\}_{t=1}^T$ used for adding noise is identical to that in Stable Diffusion. The number N of denoising steps is set to 300. For the update of codebook, we use the clustering strategy proposed in CVQ-VAE (Zheng & Vedaldi, 2023).

For training, we use the Adam (Kingma & Ba, 2014) optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for a total of 300K iterations. To achieve different compression ratios, we train five models with λ_r selected from $\{2, 1, 0.5, 0.25, 0.1\}$. The batch size is set to 4. As described in Section 3.3, the training process is divided into two stages. 1) *Independent training*. During this stage, the initial learning rate is set to 1×10^{-4} and images are randomly cropped to 512×512 patches. We first train the proposed RDEIC with $\lambda_r = 2$ for 100K iterations. The learning rate is then reduced to 2×10^{-5} and the model is trained with target λ_r for another 100K iterations. 2) *Fixed-step fine-tuning*. In this stage, the learning rate is set to 2×10^{-5} and images are randomly cropped to 256×256 patches. We fine-tune the model through the entire reconstruction process for 100K iterations. When $\lambda_r \in \{2, 1\}$, the fixed number L is set to 2, otherwise, it is 5. All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU.

C FURTHER ABLATION EXPERIMENTS

Robustness and generalization ability. To assess the robustness and generalization ability of RDEIC, we conducted additional experiments on the larger MS-COCO 30k dataset, which comprises 30,000 images spanning a diverse range of categories and content types. This dataset was constructed by selecting the same images from the COCO2017 training set (Caesar et al., 2018) as Careil et al. (2024).

As shown in Fig. 9, RDEIC maintains consistent performance across this expanded dataset, demonstrating its ability to generalize effectively to unseen data, even in scenarios with more diverse and challenging content. Visualized examples of reconstructed images are provided in Fig. 16 to further illustrate the robustness of our approach.

Role of the diffusion mechanism. To further investigate the role of the diffusion mechanism in RDEIC, we design two variants for comparison: 1) **W/o denoising process**: In this variant, the compression module is trained jointly with the noise estimator, but the denoising process is bypassed during the inference phase. 2) **W/o diffusion mechanism**: In this variant, the compression module is trained independently, completely excluding the influence of the diffusion mechanism.

As shown in Fig. 10, bypassing the denoising process results in significant degradation, particularly in perceptual quality. This demonstrates that the diffusion mechanism plays a crucial role in enhancing perceptual quality during reconstruction. As shown in Fig. 11, the diffusion mechanism effectively adds realistic and visually pleasing details.

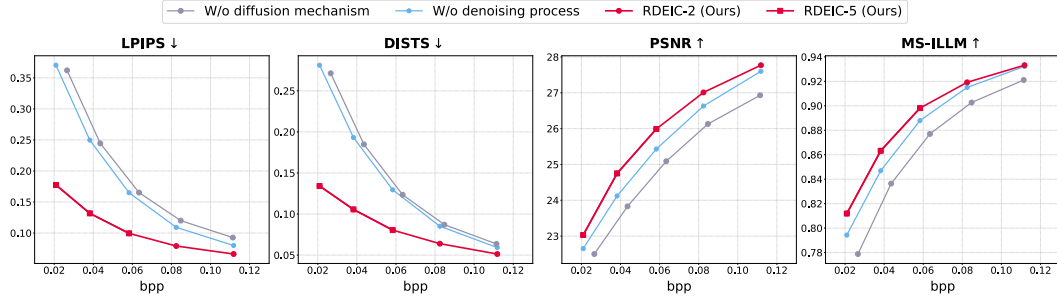


Figure 10: Ablation studies of the diffusion mechanism on CLIC2020 dataset. In the **W/o denoising process** setting, we train the compression module jointly with the noise estimator but bypass the denoising process during inference. In the **W/o diffusion mechanism** setting, we train the compression module independently, completely excluding the influence of the diffusion mechanism.



Figure 11: Impact of diffusion mechanism on reconstruction results.

By comparing the performance of **W/o diffusion mechanism** and **W/o denoising process** in Fig. 10 and Fig. 11, we observe that the compression module trained jointly with the noise estimator outperforms the one trained independently. This demonstrates that the diffusion mechanism also contributes to the compression module. Moreover, Fig. 12(a) visualizes an example of bit allocation. It is evident that the model trained jointly with the noise estimator allocates bits more efficiently, assigning fewer bits to flat regions (e.g., the sky in the image). Fig. 12(b) visualizes the cross-correlation between each spatial pixel in $(y - \mu)/\sigma$ and its surrounding positions. Specifically, the value at position (i, j) represents cross-correlation between spatial locations (x, y) and $(x + i, y + j)$ along the channel dimension, averaged across all images on Kodak dataset. It is evident that the model trained jointly with the noise estimator exhibits lower latent correlation, suggesting reduced redundancy and more compact feature representations. These results indicate that the diffusion mechanism provides additional guidance for optimizing the compression module during training, enabling it to learn more efficient and compact feature representations.

D ADDITIONAL EXPERIMENTAL RESULTS

BD-rate (%) on the CLIC2020 dataset. To provide a more intuitive comparison of overall performance on CLIC2020 dataset, we set DiffEIC (Li et al., 2024b) as the anchor and compute the

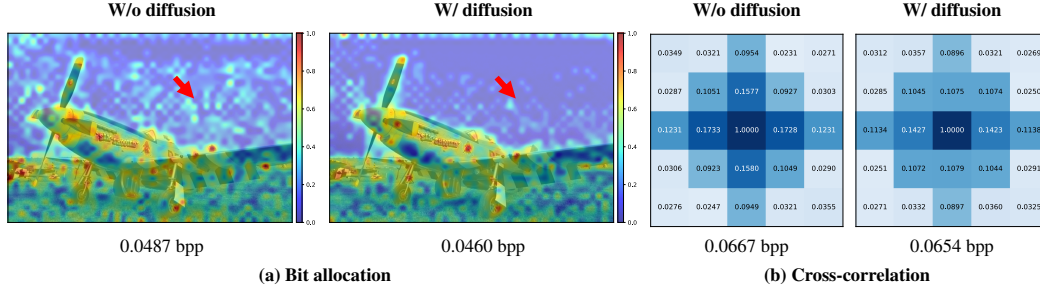


Figure 12: Impact of the diffusion mechanism on the compression module. **W/o diffusion** denotes the compression module trained independently, while **W/ diffusion** denotes the compression module trained jointly with the noise estimator. All results are obtained from models trained with $\lambda_r = 0.5$. (a) An example of bit allocation on the Kodak dataset, with the values normalized for consistency. (b) Latent correlation of $(y - \mu)/\sigma$.

Table 3: BD-rate (%) for different methods on the CLIC2020 dataset with DiffeIC as the anchor. For distortion-oriented methods (i.e., BPG, VVC, and ELIC), we omit their perceptual metrics. The best and second best results are highlighted in **bold** and underline.

Methods	Perception					Distortion			Average
	DISTS	FID	KID	NIQE	LPIPS	PSNR	MS-SSIM	SSIM	
BPG	—	—	—	—	—	-66.2	-32.8	-40.3	—
VVC	—	—	—	—	—	<u>-77.8</u>	<u>-51.3</u>	<u>-58.6</u>	—
ELIC	—	—	—	—	—	-82.7	-54.6	-66.7	—
HiFiC	201.8	248.2	372.6	-28.7	63.4	-29.1	2.7	14.7	105.7
VQIR	71.8	183.9	156.7	32.4	51.3	16.4	43.9	57.8	76.8
PerCo	66.1	67.6	65.1	5.2	67.7	33.9	69.2	77.7	56.6
MS-ILLM	<u>28.5</u>	<u>40.9</u>	<u>34.6</u>	-85.4	-44.7	-75.4	-44.7	-38.5	<u>-21.5</u>
RDEIC(Ours)	-17.9	-18.3	-22.1	<u>-83.7</u>	<u>-40.8</u>	-61.3	-32.7	-32.7	-38.7

BD-rate (Bjontegaard, 2001) for each metric. As shown in Table 3, our method outperforms all perception-oriented comparison methods, achieving the lowest average BD-rate value among them.

Quantitative comparisons on the Tecnick and Kodak datasets. We present the performance of the proposed and compared methods on the Tecnick and Kodak datasets in Fig. 14 and Fig. 15, respectively. The proposed RDEIC achieves state-of-the-art perceptual performance and significantly outperforms other diffusion-based methods in terms of distortion metrics. Since the Kodak dataset is too small to reliably calculate FID and KID scores, we do not report these results for this dataset.

Smoothness-sharpness trade-off. As shown in Fig. 17, Fig. 18, and Fig. 19, we control the balance between smoothness and sharpness by adjusting the parameter λ_s , which regulates the amount of high-frequency details introduced into the reconstructed image.

E LIMITATIONS

Using pre-trained stable diffusion may generate hallucinated lower-level details at extremely low bitrates. For instance, as shown in Fig. 13, the generated human faces appear realistic but are inaccurate, which may lead to a misrepresentation of the person’s identity. Furthermore, although the proposed RDEIC has shown promising compression results, the potential of incorporating a text-driven strategy has not yet been explored within our framework. We leave detailed study of this to future work.

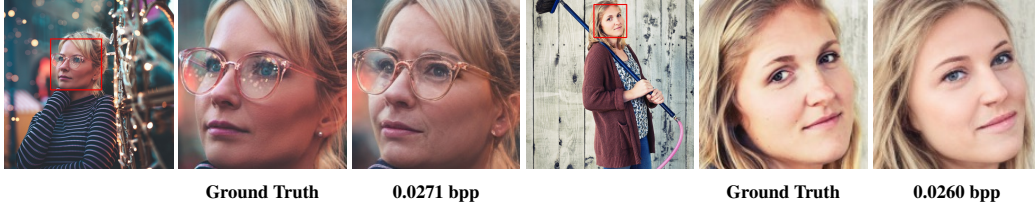


Figure 13: Faces generated at extremely low bitrates.

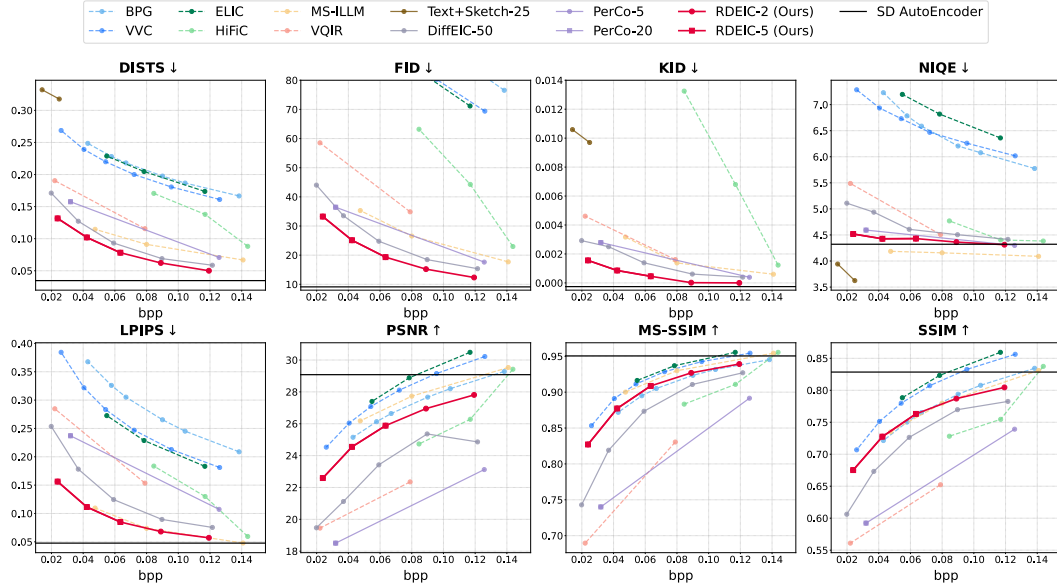


Figure 14: Quantitative comparisons with state-of-the-art methods on the Tecnick dataset.

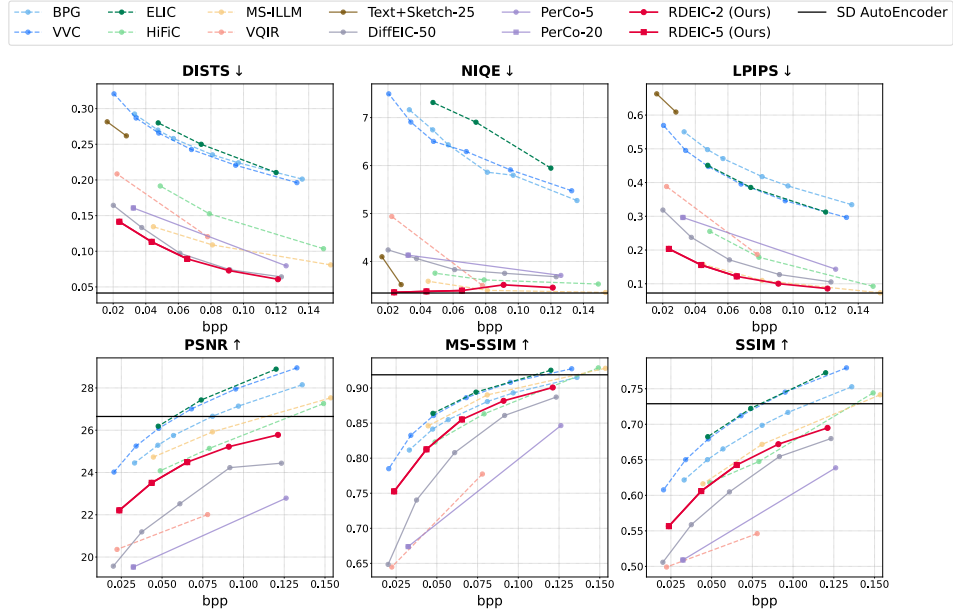


Figure 15: Quantitative comparisons with state-of-the-art methods on the Kodak dataset.

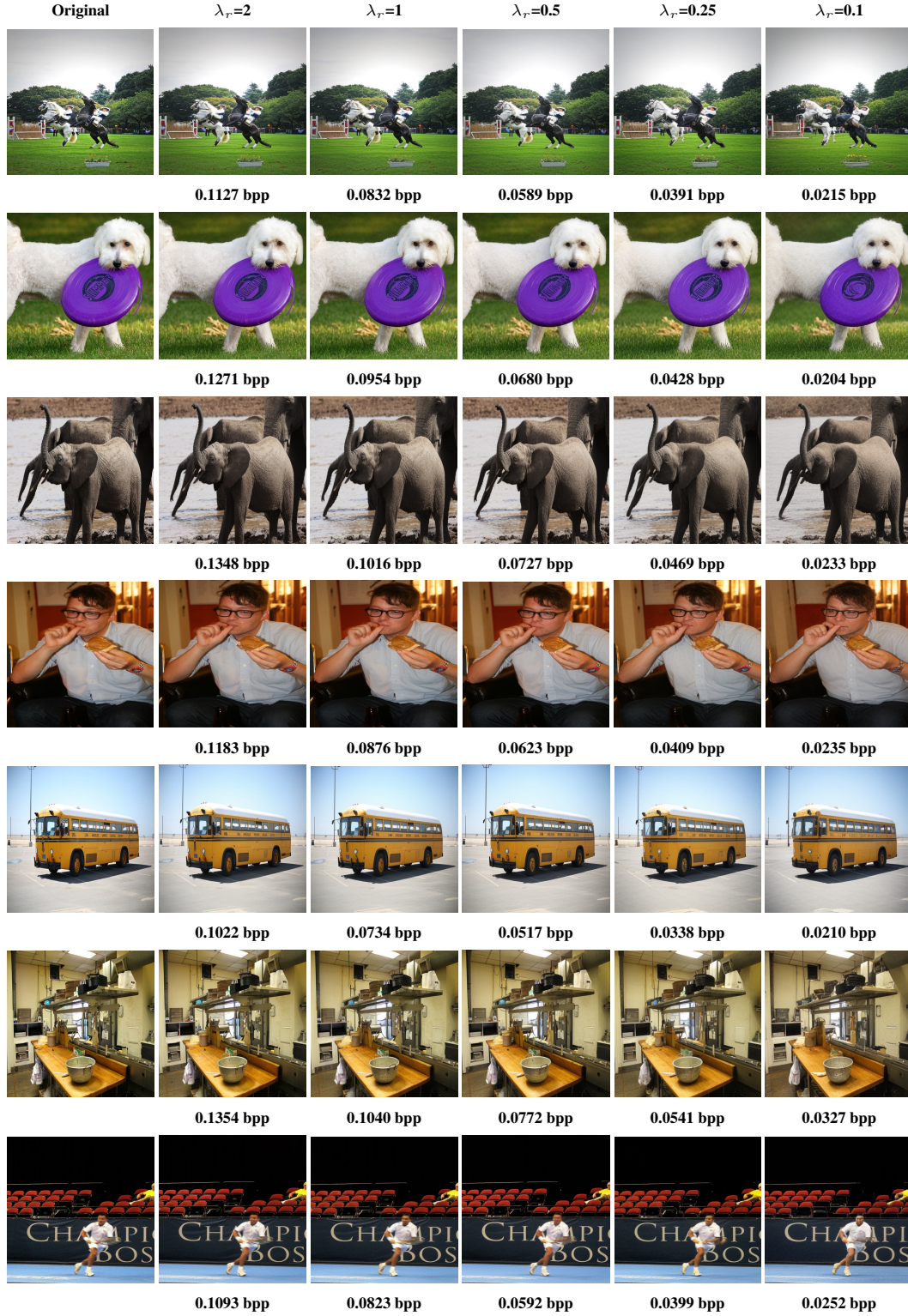


Figure 16: Visualization results of RDEIC on the MS-COCO 30k dataset at different bitrates.

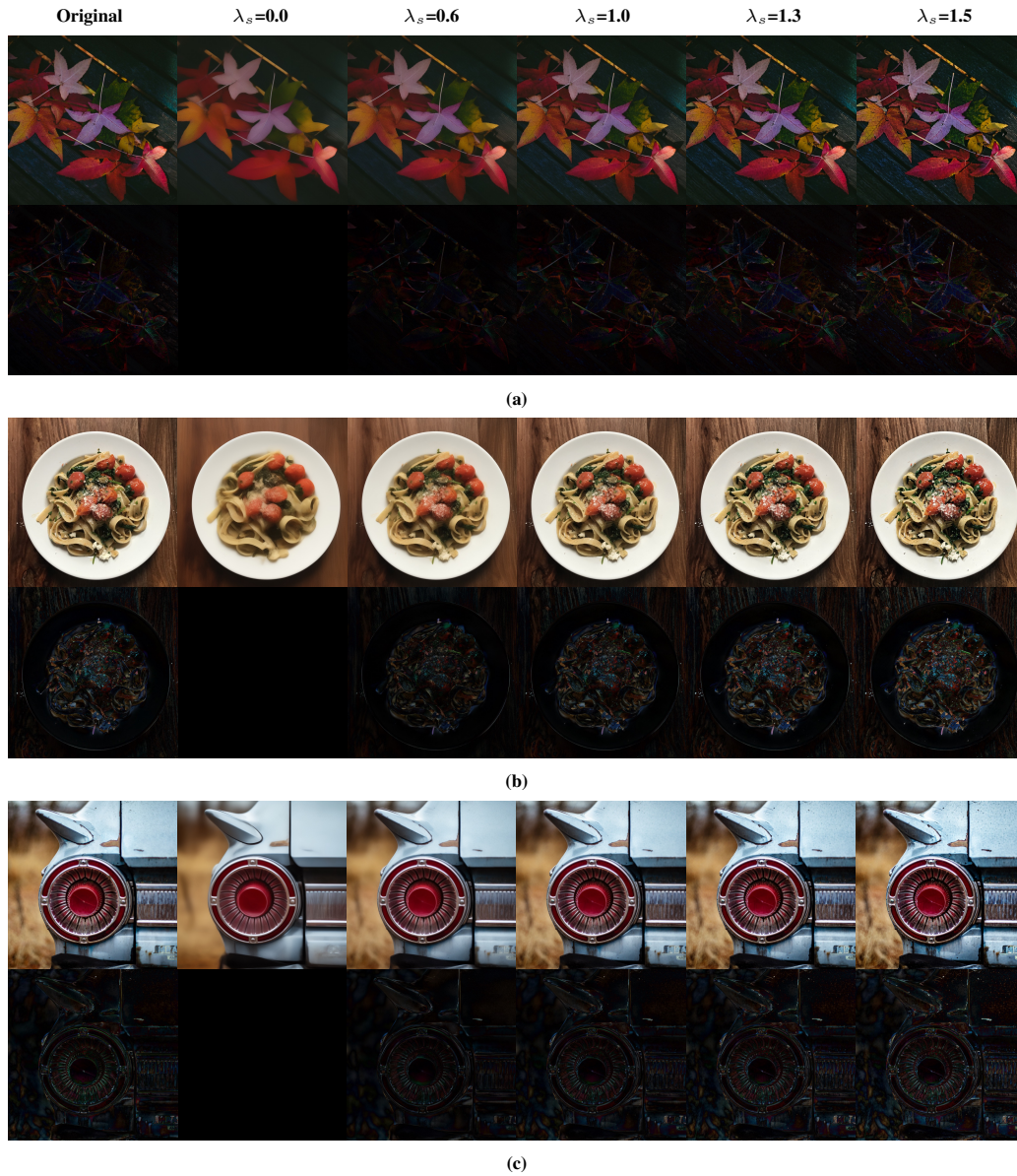


Figure 17: More results regarding the balance between smoothness and sharpness.



Figure 18: More results regarding the balance between smoothness and sharpness.



Figure 19: More results regarding the balance between smoothness and sharpness.