StableCodec: Taming One-Step Diffusion for Extreme Image Compression

Supplementary Material

A. Inference for Arbitrary Resolution

Diffusion models typically face scalability issues when dealing with high-resolution images, often yielding inferior results while incurring significantly increased computational costs. Consequently, existing diffusion-based codecs [3, 13, 16] primarily target small images with resolutions around 512×512 or resized images. To enhance the practicality of StableCodec, we adopt a tiled VAE approach [1] to split high-resolution images into tiles and process them sequentially in both the VAE encoder and decoder. For one-step denoising, we employ a similar latent aggregation strategy [9, 22], which processes latent patches individually and aggregates overlapping pixels using a Gaussian weight map. These methods enable StableCodec to support arbitrary-resolution inference with memory consumption under 9 GB, greatly improving its efficiency and practicality for real-world deployment.

However, we observe that StableCodec sometimes produces color shifts when reconstructing high-resolution images, as illustrated in Fig. 8. This issue has also been noted in [4, 22]. To address this, we apply a quantized version of adaptive instance normalization [22] on the reconstructed high-resolution image $\hat{x}$, aligning its mean ($\mu_{\hat{x}}$) and variance ($\sigma_{\hat{x}}$) with those of the original image (μ_x and σ_x):

$$\hat{x}^c = \frac{\hat{x} - \mu_{\hat{x}}}{\sigma_{\hat{x}}} \cdot \hat{\sigma_x} + \hat{\mu_x} \tag{9}$$

where $\hat{\mu_x}$ and $\hat{\sigma_x}$ are 16-bit-quantized from μ_x and σ_x:

$$\hat{\mu_x} = \frac{\lfloor \mu_x \cdot (2^{16} - 1) + 2^{-1} \rfloor}{2^{16} - 1} \tag{10}$$

$$\hat{\sigma_x} = \frac{\lfloor \sigma_x \cdot (2^{16} - 1) + 2^{-1} \rfloor}{2^{16} - 1} \tag{11}$$

Here, $\hat{x}^c$ represents the color-corrected reconstruction, and μ_x and σ_x contain the mean and variance values for the RGB channels, each represented as 32-bit floating point values. We find that quantizing these values to 16 bits does not significantly affect correction performance. This strategy effectively refines the color of high-resolution reconstructions with only a minimal increase in bit cost (96 bits per image), as demonstrated in Fig. 8.

B. Network Structure

We present our entropy model in Fig. 9, with the detailed network architecture shown in Fig. 10. Given the quantized latent $\hat{y}$, the entropy model estimates its distribution for arithmetic coding. Following [19], our entropy model is built with a hyperprior module and an autoregressive context model, where we first obtain and transmit a hyperprior Φ_{hyper} from y using the hyper transform h_a and h_s:

$$z = h_a(y), \hat{z} = Q(z), \Phi_{hyper} = h_s(\hat{z}) \tag{12}$$

Here, y has 320 channels with 64× (a spatial compression ratio of 64), while z and $\hat{z}$ have 160 channels with 256×. To balance the coding performance and efficiency, we construct a 4-step autoregressive process using quadtree partition [14] and latent residual prediction [18]. The detailed autoregressive process to estimate the Gaussian parameters, μ and σ, for $\hat{y}$ is illustrated in Fig. 9. Following this, arithmetic coding is applied to encode $\hat{y}$ into a bitstream, or decode $\hat{y}$ from the bitstream. For efficient network construction, we primarily rely on modified versions of InceptionNeXt [24] and GatedCNN [23], as detailed in Fig. 10.

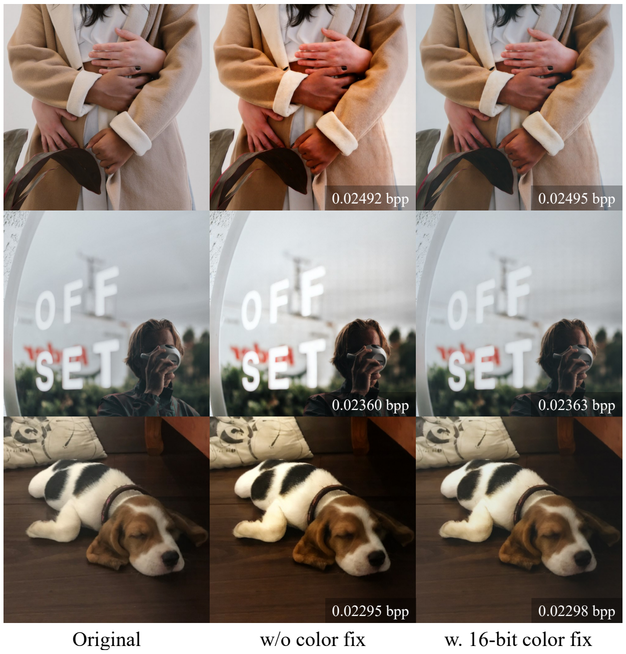

Figure 8. **Visual examples of color fix** from CLIC 2020 [21]. 16-bit color fix brings clear refinement with negligible bits increase.

C. Runtime Analysis

We conduct detailed runtime analysis of different modules in StableCodec using a single RTX 3090 GPU, and display the results in Table 6. Specifically, we examine the time consumption of the VAE encoder $\mathcal{E}_{\text{SD}}$, auxiliary encoder $\mathcal{E}_{\text{Aux}}$, g_a and entropy encoding during the encoding

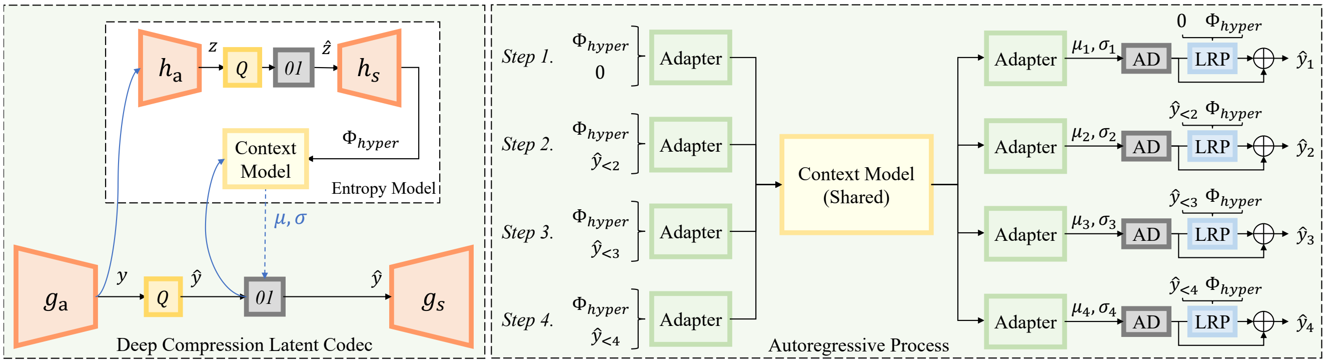

Figure 9. **(Left) Illustration of the entropy model.** We build our entropy model on the basis of [19], which consists of a pair of hyper transforms, h_a and h_s, and a context model to perform entropy estimation for $\hat{y}$ in an autoregressive manner. **(Right) Illustration of the 4-step autoregressive process.** We divide $\hat{y}$ into 4 groups ($\hat{y}_1$, $\hat{y}_2$, $\hat{y}_3$ and $\hat{y}_4$) using quadtree partition [14]. For each $\hat{y}_i$, we estimate its Gaussian parameters, μ_i and σ_i, with the hyperprior Φ_{hyper} and previously decoded groups $\hat{y}_{<i}$. The parameter networks contain a shared context model and private adapters. AD represents arithmetic decoding the bitstream of $\hat{y}_i$ given corresponding Gaussian parameters, μ_i and σ_i. Additionally, we incorporate latent residual prediction (LRP) [18] to alleviate the quantization error.

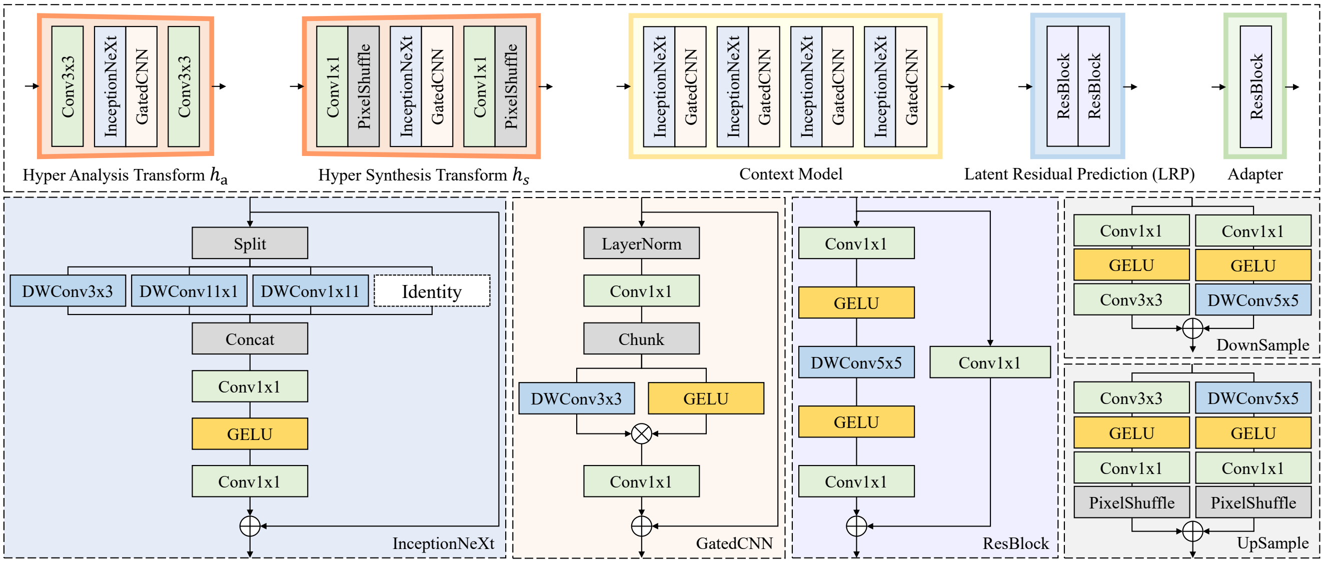

Figure 10. **Module structures and network details.**

Table 6. **Runtime analysis of specific modules in seconds** averaged on Kodak [6]. $\mathcal{E}_{\text{SD}}$ and $\mathcal{D}_{\text{SD}}$ represent the VAE encoder and decoder of SD-Turbo, while EE and ED denote entropy encoding and decoding with the entropy model. We add representative neural codec ELIC [7] for comparison, which only contains the analysis transform g_a, the synthesis transform g_s and the entropy model.

Method	Encoding Time (s)				Decoding Time (s)				
	$\mathcal{E}_{\text{SD}}$	$\mathcal{E}_{\text{Aux}}$	g_a	EE	ED	g_s	$\mathcal{D}_{\text{Aux}}$	ϵ_{SD}	$\mathcal{D}_{\text{SD}}$
StableCodec (Ours)	0.108	0.014	0.005	0.029	0.041	0.004	0.004	0.112	0.161
ELIC [7]	-	-	0.015	0.138	0.230	0.016	-	-	-

process, and those of the entropy decoding, g_s, auxiliary decoder $\mathcal{D}_{\text{Aux}}$, one-step denoising Unet ϵ_{SD} and VAE decoder $\mathcal{D}_{\text{SD}}$ during the decoding process. For comparison, we add the representative VAE-based neural codec ELIC [7], which only contains g_a, g_s and the entropy model.

Since we use the analysis transform g_a of a pre-trained ELIC model to serve as $\mathcal{E}_{\text{Aux}}$, the time consumption of "StableCodec - $\mathcal{E}_{\text{Aux}}$" is close to that of "ELIC - g_a". Besides, the time consumption of g_a, g_s and entropy coding in StableCodec is much smaller than those of ELIC. This is be-

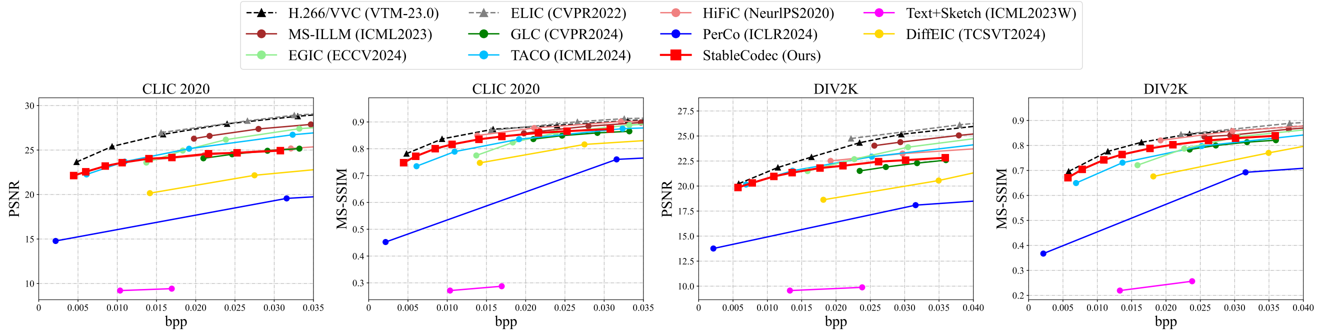

Figure 11. **Additional rate-distortion curves on CLIC 2020 [21] and DIV2K [2] in terms of PSNR and MS-SSIM.**

Table 7. **Top-1 user preference.** We evaluate reconstructions from different methods at similar ultra-low bitrates using the Kodak dataset [6]. Our study involves 30 participants, yielding a total of 720 evaluated cases. In each case, we display the ground-truth image alongside eight reconstructions from different methods, and invite participates to select the most "consistent" one compared with the ground-truth.

Method	HiFiC	MS-ILLM	Text+Sketch	PerCo	DiffEIC	EGIC	TACO	StableCodec (Ours)
Bitrate (bpp)	0.0268	0.0262	0.0274	0.0321	0.0375	0.0247	0.0258	0.0250
Top-1 Votes	20	26	11	24	43	29	54	**513**
Percentage	2.78%	3.61%	1.53%	3.33%	5.97%	4.03%	7.50%	**71.25%**

cause StableCodec adopts Deep Compression Latent Codec with advanced 4-step autoregressive entropy model and network designs, performing efficient transform coding at 16× and entropy estimation at 64×, while ELIC performs transform coding on original images and entropy estimation at 16×. Benefit from these designs, StableCodec is able to achieve comparable coding speed with mainstream neural codecs, significantly outperforms existing diffusion-based methods as suggested in Table 2.

D. User Study

To provide a more comprehensive evaluation of reconstruction quality at ultra-low bitrates, we conduct a user study on the Kodak dataset [6] using a top-1 user preference approach. We compare StableCodec against seven representative generative image codecs: HiFiC [17], MS-ILLM [20], Text+Sketch [13], PerCo [3], DiffEIC [16], EGIC [10], and TACO [12], all evaluated at similar average bitrates. To produce the reconstructions, we use the official weights of Text+Sketch, PerCo (SD) [11] and DiffEIC, while HiFiC, MS-ILLM, EGIC and TACO are either re-trained or fine-tuned from existing weights to reach specific bitrates.

Each participant in our study examines 24 cases, requiring an average of three minutes to complete. For each case, we present a ground-truth image alongside eight reconstructions from different methods, displayed in 2 rows and 4 columns with random order. Participants are asked to select the reconstruction they find most "consistent" with the ground-truth image. A total of 30 participants completed the study, yielding 720 evaluated cases. The results, summarized in Table 7, show that StableCodec reconstructions were preferred in over 70% of cases, demonstrating its superior visual consistency as perceived by human observers.

E. Visual Performance

In this section, we display more visual examples and comparisons on high-quality images from DIV2K [2] (Fig. 12), CLIC 2020 [21] (Fig. 13) and USTC-TD [15] (Fig. 14 and Fig. 15). We compare the proposed StableCodec with existing methods, including ELIC [7], MS-ILLM [20], PerCo [3], EGIC [10], DiffEIC [16], and TACO [12], all at ultra-low bitrates. Notably, StableCodec outperforms the competing methods in terms of both semantic consistency and textual realism, while consuming fewer bits.

F. Quantitative Results

In Fig. 11, we provide additional PSNR and MS-SSIM comparisons on CLIC 2020 and DIV2K as a supplement for Fig. 6. As discussed in Section 4.1, pixel-level metrics like PSNR, MS-SSIM, and LPIPS have notable limitations [3, 5, 8, 13] due to their emphasis on pixel accuracy rather than semantic consistency or textual realism, making them less suitable for evaluating ultra-low bitrate compression. Therefore, for StableCodec, we primarily focus on FID, KID, and DISTS, which offer a more accurate assessment of quality in severely compressed images.

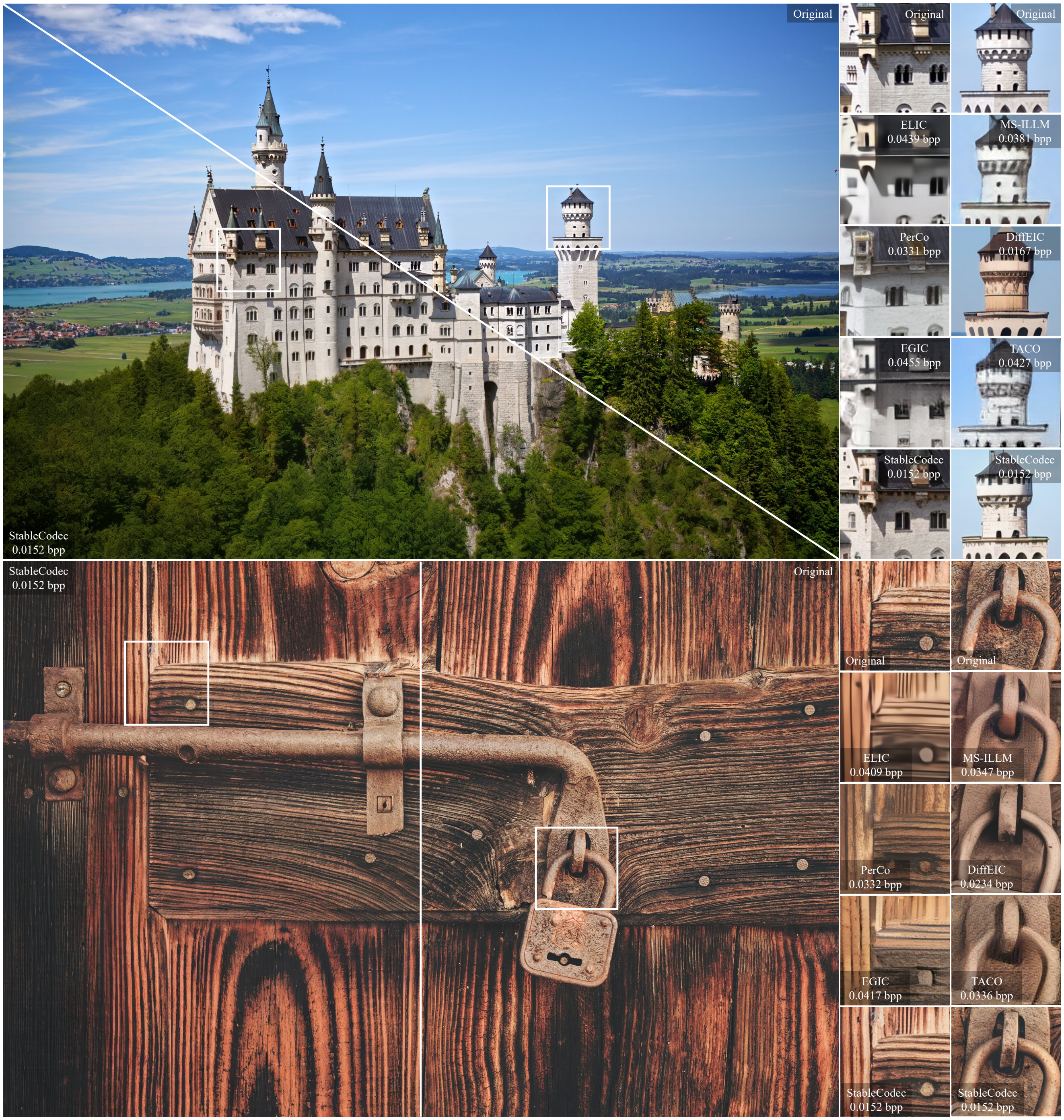

Figure 12. **Visual examples and comparisons on 2K-resolution images from DIV2K.**

References

[1] Tiled diffusion & vae extension. https://github.com/pkuliyi2015/multidiffusion-upscaler-for-automatic1111, 2023. Accessed: 2024-08-27. 1

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 3

[3] Marlene Careil, Matthew J Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *The Twelfth International Conference on Learning Representations*, 2023. 1, 3

[4] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon

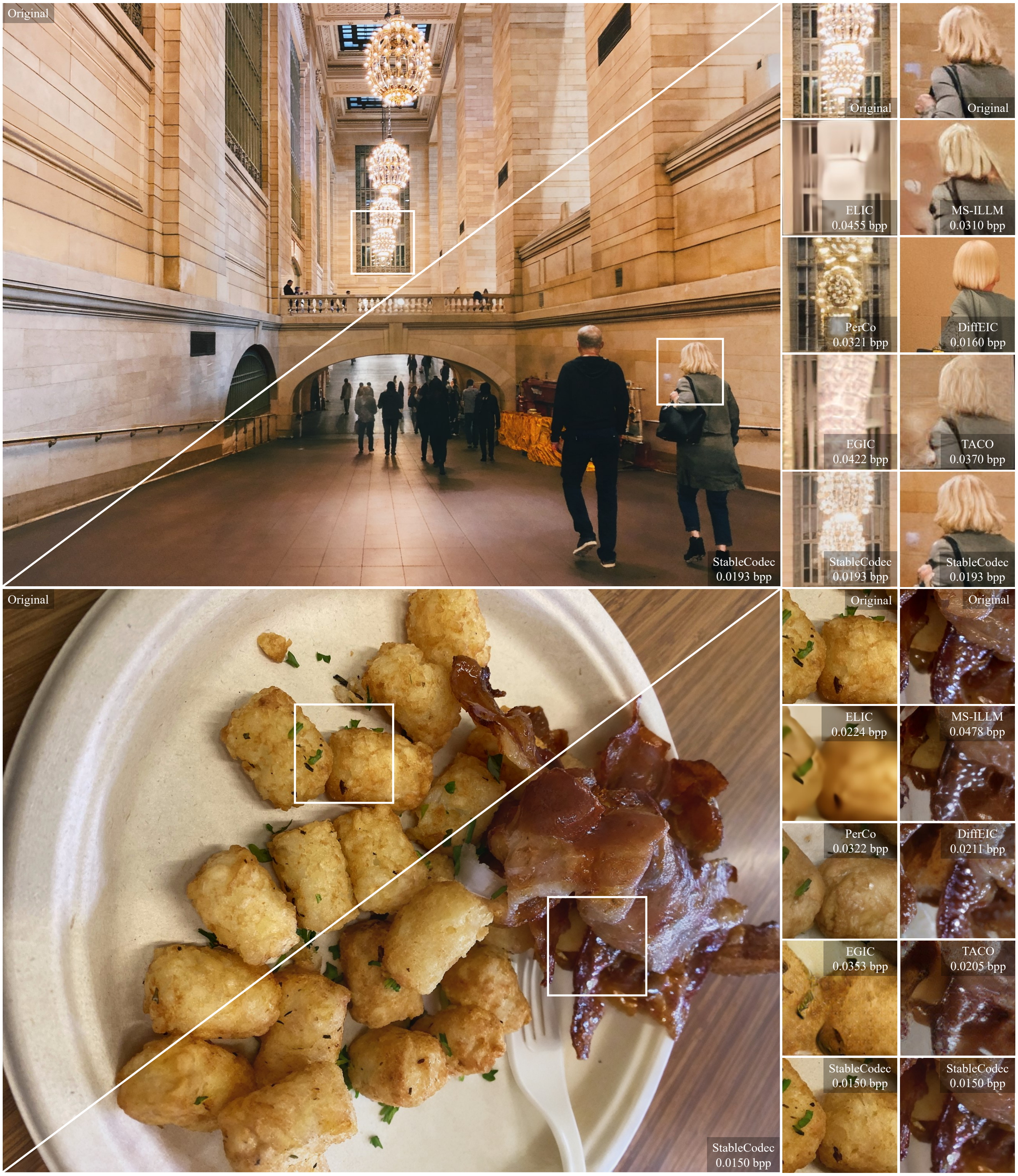

Figure 13. **Visual examples and comparisons on 2K-resolution images from CLIC 2020.**

Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022. 1

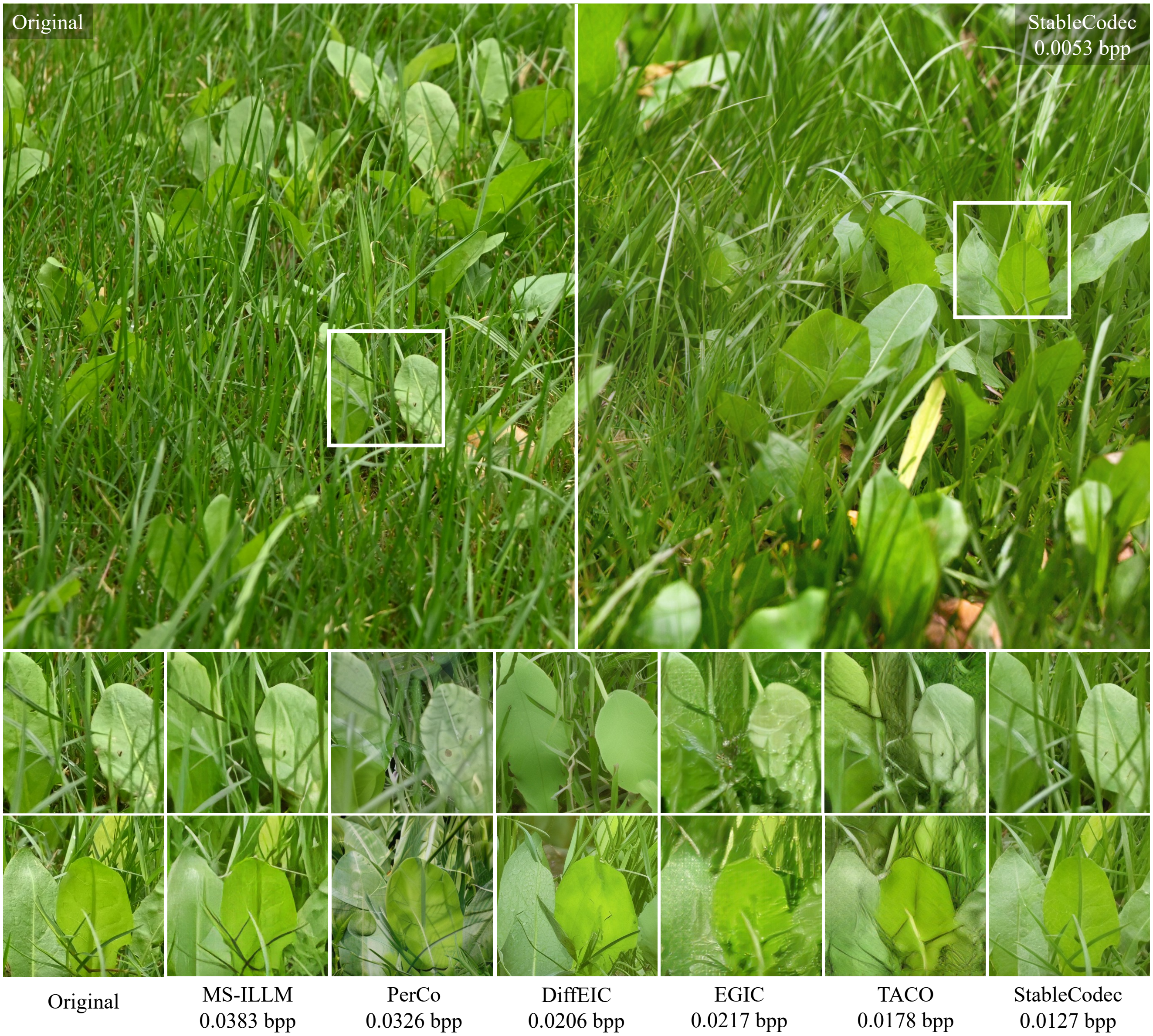

Figure 14. **Visual examples and comparisons on 4K-resolution images from USTC-TD [15].**

[5] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020. 3

[6] Rich Franzen. Kodak lossless true color image suite (photocd pcd0992). `http://r0k.us/graphics/kodak/`, 1993. 2, 3

[7] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5718–5727, 2022. 2, 3

[8] Zhaoyang Jia, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Generative latent coding for ultra-low bitrate image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26088–26098, 2024. 3

[9] Álvaro Barbero Jiménez. Mixture of diffusers for scene composition and high resolution image generation. *arXiv preprint arXiv:2302.02412*, 2023. 1

[10] Nikolai Körber, Eduard Kromer, Andreas Siebert, Sascha Hauke, Daniel Mueller-Gritschneder, and Björn Schuller. Egic: enhanced low-bit-rate generative image compression guided by semantic segmentation. In *European Conference on Computer Vision*, pages 202–220. Springer, 2024. 3

[11] Nikolai Körber, Eduard Kromer, Andreas Siebert, Sascha Hauke, Daniel Mueller-Gritschneder, and Björn Schuller.

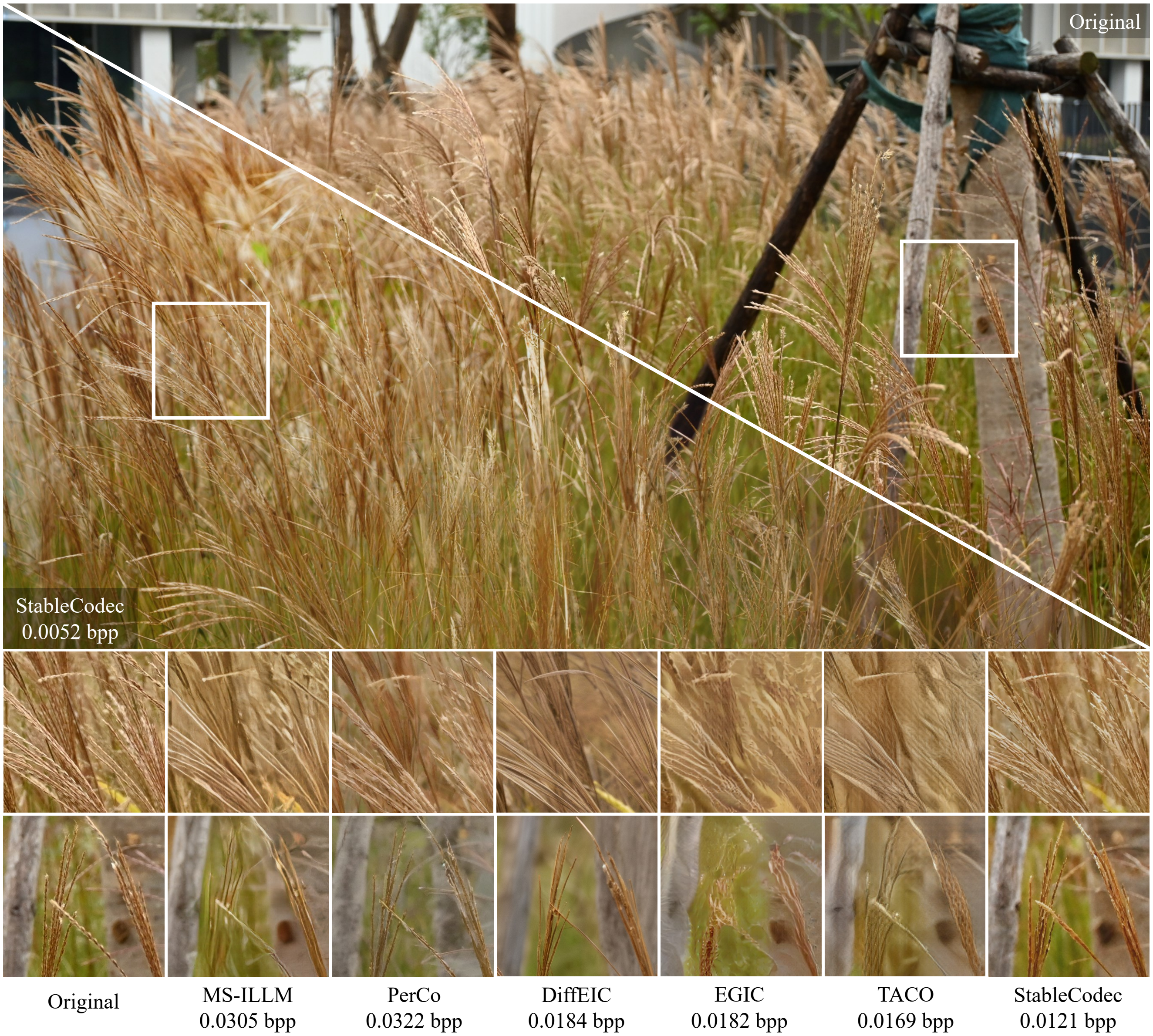

Figure 15. **Visual examples and comparisons on 4K-resolution images from USTC-TD [15].**

Perco (sd): Open perceptual compression. *arXiv preprint arXiv:2409.20255*, 2024. 3

[12] Hagyeong Lee, Minkyu Kim, Jun-Hyuk Kim, Seungeon Kim, Dokwan Oh, and Jaeho Lee. Neural image compression with text-guided encoding for both pixel-level and perceptual fidelity. *arXiv preprint arXiv:2403.02944*, 2024. 3

[13] Eric Lei, Yiğit Berkay Uslu, Hamed Hassani, and Shirin Saeedi Bidokhti. Text+ sketch: Image compression at ultra low rates. *arXiv preprint arXiv:2307.01944*, 2023. 1, 3

[14] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22616–22626, 2023. 1, 2

[15] Zhuoyuan Li, Junqi Liao, Chuanbo Tang, Haotian Zhang, Yuqi Li, Yifan Bian, Xihua Sheng, Xinmin Feng, Yao Li, Changsheng Gao, et al. Ustc-td: A test dataset and benchmark for image and video coding in 2020s. *arXiv preprint arXiv:2409.08481*, 2024. 3, 6, 7

[16] Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Jingwen Jiang. Towards extreme image compression with latent feature guidance and diffusion prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 1, 3

[17] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33:11913–11924, 2020. 3

[18] David Minnen and Saurabh Singh. Channel-wise autoregres-

sive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2020. 1, 2

[19] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018. 1, 2

[20] Matthew J Muckley, Alaaeldin El-Nouby, Karen Ullrich, Hervé Jégou, and Jakob Verbeek. Improving statistical fidelity for neural image compression with implicit local likelihood models. In *International Conference on Machine Learning*, pages 25426–25443. PMLR, 2023. 3

[21] George Toderici, Lucas Theis, Nick Johnston, Eirikur Agustsson, Fabian Mentzer, Johannes Ballé, Wenzhe Shi, and Radu Timofte. Clic 2020: Challenge on learned image compression, 2020, 2020. 1, 3

[22] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *International Journal of Computer Vision*, pages 1–21, 2024. 1

[23] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? *arXiv preprint arXiv:2405.07992*, 2024. 1

[24] Weihao Yu, Pan Zhou, Shuicheng Yan, and Xinchao Wang. Inceptionnext: When inception meets convnext. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5672–5683, 2024. 1