
Supplementary Material of One-Step Diffusion-Based Image Compression with Semantic Distillation

A Experiment

Evaluation of third-party models. We evaluate MS-ILLM [14] using the official checkpoints and fine-tune them with the provided code to support lower bitrates. For GLC [6], we report the results directly from its paper because the code and models are not publicly available at present. For DiffEIC [9] and DiffC [17], we use the official implementations and released models. Specifically, we adopt the Stable Diffusion (SD) 2.1-based variant of DiffC, which shows slightly better performance compared to the SD 1.5 version. For PerCo [3], we use a publicly available reimplementation PerCo (SD) [8], as the original code has not been released. We also report the comparison with original PerCo by extract data from their paper (Fig. A). For DDCM [15], we use the number reported in their paper. For multi-step diffusion codecs, we follow the default sampling settings provided in their code (e.g., 50 for DiffEIC, 20 for PerCo).

Test settings and FID calculation. At the *full-resolution* setting, we compute FID using overlapping 256×256 patches for the CLIC2020 and DIV2K test sets [1], following the protocol of [13]. For the MS-COCO 30K dataset with 512×512 images, FID is evaluated on entire images, consistent with [3, 8]. At the *resize & center-crop* setting, we resize the short side of each image (512 for Kodak, 768 for CLIC2020 test set) and then apply a center crop. In this setting, we use 64×64 patches for FID calculation on Kodak and 128×128 patches on CLIC2020 test set, consistent with DDCM [15]. It is worth noting that DiffEIC also evaluates under the *resize & center-crop* setting on CLIC2020 test set at 768 resolution, but computes FID with overlapping 256×256 patches. For completeness, we additionally report results under DiffEIC’s protocol, denoted as *resize & center-crop, 256 FID*.

Additional datasets and metrics. To enable comprehensive comparison, we further evaluate our method on the DIV2K test set [1] under the *full-resolution* setting, as shown in Fig. B. We also report results on the CLIC2020 test set at 768 resolution under DiffEIC’s protocol, i.e., *resize & center-crop, 256 FID*, as illustrated in Fig. C. Across datasets, metrics, and evaluation settings, the OneDC consistently achieves SOTA performance, demonstrating strong robustness and generalization. The raw data used for evaluation is available at Table. C ~ H.

We also report traditional pixel-level distortion metrics (PSNR and MS-SSIM [18]) to provide a more comprehensive analysis, as shown in Fig. D. At extremely low bitrates, optimizing for PSNR often suppresses high-frequency details, resulting in blurred reconstructions [13]. While OneDC shows slightly lower PSNR than MS-ILLM, both perceptual metrics and qualitative examples clearly demonstrate its superior visual quality. Despite prioritizing perceptual quality, OneDC still achieves competitive PSNR compared to other diffusion-based methods. Moreover, on the MS-SSIM metric, OneDC matches MS-ILLM and outperforms all other baselines. These results confirm that OneDC delivers strong pixel-level fidelity alongside high perceptual realism, highlighting the overall effectiveness of our method.

Additional visual examples. We present more qualitative comparisons across four datasets: Kodak (Fig.G), CLIC2020—both full resolution (Figs.H) and 768 \times 768 cropped (Fig.I), and MS-COCO 30K (Fig.J). OneDC consistently outperforms prior SOTA methods, delivering superior visual quality across diverse content and resolutions, yet with the lowest bitrate cost.

Effectiveness of semantic distillation. To further evaluate the proposed semantic distillation strategy, we fine-tune the pretrained text-to-image one-step diffusion model [19], replacing its textual semantic condition with the hyperprior features produced by our semantic hyperprior decoder h_{sem} . This

Table A: Complexity analysis with model size. BD-Rate is calculated on the MS-COCO 30K dataset.

Model	Params	Enc. Time (s)	Dec. Time (s)	BD-Rate (Metircs)↓		
				LPIPS	DISTS	FID
MS-ILLM	181M	0.14	0.17	138.3%	253.0%	478.4%
DiffEIC	1.4B	0.32	12.4	305.0%	239.1%	341.0%
PerCo (SD)	3.8B+340M+955M*	0.58	8.80	538.8%	345.8%	59.6%
DiffC	950M	3.9~15.6	6.9~10.8	234.0%	196.1%	690.9%
DDCM	950M	-	-	-	-	-
OneDC	1.4B	0.15	0.34	0.00%	0.00%	0.00%

* Open-sourced PerCo includes an additional 3.8B BLIP2 caption model and 340M CLIP model.

Table B: Bitrate allocation analysis on the CLIC2020 dataset.

Bpp \hat{z} (ratio)	Bpp \hat{y} (ratio)	Bpp Total	PSNR↑	MS-SSIM↑	LPIPS↓	DISTS↓	FID↓
0.0035 (100%)	0.0 (0%)	0.0035	19.31	0.629	0.290	0.169	14.885
0.0035 (43%)	0.0047 (57%)	0.0082	23.13	0.790	0.163	0.089	6.223
0.0035 (27%)	0.0094 (73%)	0.0129	24.20	0.826	0.139	0.077	5.560
0.0035 (18%)	0.0157 (82%)	0.0192	25.25	0.856	0.119	0.068	4.979

enables reconstruction using only the hyperprior signal. Fig. K presents additional reconstruction results on the COCO2017 validation dataset [10], extending Fig. 2 from the main paper. These results confirm that the distilled model captures richer semantic information, facilitating final reconstruction.

Bitrate allocation between hyperprior \hat{z} and latent \hat{z} . We ablate the roles of hyperprior \hat{z} and latent \hat{y} on CLIC2020 test set by fixing the hyperprior budget to 0.0035 bpp and gradually increasing the bitrate of \hat{y} . Table B reports the results and reveals a clear division of each part’s role: (a) with zero bits allocated for latent \hat{y} , the codec still produces coherent reconstructions (also show in Fig. K), indicating that \hat{z} supplies a strong semantic information; (b) as bits are assigned to \hat{y} , fidelity and perceptual quality improve monotonically (e.g., LPIPS drops from 0.290 to 0.119), confirming that \hat{y} is essential for encoding fine-grained details.

Model size. We provide parameter counts, runtime comparisons (on 1024×1024 images), and BD-Rate results on the MS-COCO 30K dataset for better efficiency evaluation, as shown in Table A. Compared to MS-ILLM, diffusion-based methods typically use larger models but achieve superior perceptual quality (e.g., lower BD-Rate with FID) due to stronger generative capacity. Unlike other diffusion-based codecs, OneDC avoids external caption models and multi-step sampling, enabling over 20× faster decoding while also achieving better rate-distortion performance.

Memroy usage. We also report memory usage under the *resize & center-crop* setting on 512 × 512 Kodak images. PerCo (SD) requires about 22,220 MB of GPU memory, whereas DDCM uses 4,186 MB and OneDC uses 8,038 MB. These results highlight that incorporating a large language model, as in PerCo, substantially increases computational burden. DDCM achieves lower memory consumption by employing the SD model in a zero-shot manner. Although OneDC requires more memory than DDCM, its single-step design reduces inference cost compared to both multi-step PerCo and DDCM.

B Training Details

Stage I training. This stage focuses on training the compression module and fine-tuning the one-step diffusion model [19] for the image reconstruction task. The training loss is defined as:

$$L_{stageI} = L_{recon} + \lambda R + \alpha L_{aux}, \quad \text{where } L_{recon} = L_1(x, \hat{x}) + L_{perceptual}(x, \hat{x}) \quad (1)$$

We use the L1 as the pixel-level loss and the LPIPS [7] as the perceptual-level loss. To support various bitrates, the rate-distortion trade-off parameter λ is set to $\{0.6, 1.0, 1.8, 2.9, 4.6, 7.4, 12.2\}$. An auxiliary code prediction loss L_{aux} is included with a weighting factor of $\alpha = 0.001$. We train our model on the dataset introduced in [4]. Training is performed on 4×A100 GPUs for 800,000 steps, using a three-stage learning rate schedule with AdamW [12]: a) 5e-5 for the first 500,000 steps; b) 1e-5 for the next 200,000 steps; c) 1e-6 for the final 100,000 steps. During training, image patches of size $\{512, 1024\}$ are randomly cropped with probabilities of $\{0.6, 0.4\}$, respectively. The batch

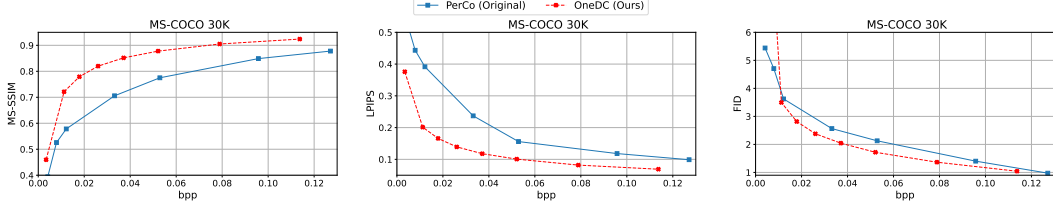


Figure A: Comparison with original PerCo [3] on the MS-COCO 30K dataset.

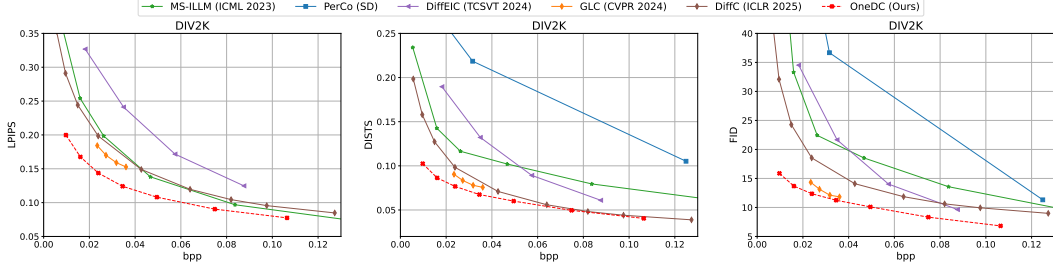


Figure B: Rate-distortion curves on the DIV2K dataset at *full-resolution* setting.

size is set to 32 for 512×512 crops and 8 for 1024×1024 crops (across 4 GPUs). This stage takes approximately 6 days as we use high-resolution patches for training.

Stage II training. This stage fine-tunes the one-step generator to improve reconstruction realism by better aligning the distribution of its outputs with that of real images. The training objective is:

$$L_{stageII} = L_{distill} + \beta L_{recon} + \gamma L_{adv}, \quad \text{where:} \quad (2)$$

$$L_{distill} = \mathbb{E}_{t, \tilde{y}'_t} [\epsilon_{fake}(\tilde{y}'_t, t) - \epsilon_{real}(\tilde{y}'_t, t)], \quad L_{adv} = \mathbb{E}_{t, \tilde{y}'_t} [-Disc(\epsilon_{fake}(\tilde{y}'_t, t), t)] \quad (3)$$

Here, $L_{distill}$ represents the diffusion distillation loss [20], and L_{adv} is the adversarial loss in the latent space, following [19]. $Disc$ denotes the discriminator network, which takes the mid feature in the diffusion U-Net as the input [19]. The variable \tilde{y}_0 is the latent output generated by the one-step diffusion model, and \tilde{y}'_t is its noised version at timestep t . We uniformly sample $t \in [20, 640]$, since synthesizing high-frequency details does not require large noise levels. The weighting parameters are set as follows: $\beta = 0.625$ balance the reconstruction and distillation terms, and $\gamma = 0.001$ for the adversarial loss (γ follows [19]).

Specifically, diffusion distillation [20] minimizes the expected Kullback-Leibler (KL) divergence between the time-dependent distributions of the target $p_{real,t}$ and the generator output $p_{fake,t}$, thereby effectively transferring knowledge from the multi-step diffusion model to the one-step generator. The gradient used to update the one-step generator parameters θ is given by the difference between the score functions of the real and fake distributions:

$$\nabla_{\theta} L_{distill} = \mathbb{E}_{t, \tilde{y}_0} (\nabla_{\theta} \text{KL}(p_{fake,t} || p_{real,t})) \quad (4)$$

$$= -\mathbb{E}_{t, \tilde{y}'_t} [(s_{real}(\tilde{y}'_t, t) - s_{fake}(\tilde{y}'_t, t)) \frac{d\epsilon_{\theta}}{d\theta}] \quad (5)$$

$$= \mathbb{E}_{t, \tilde{y}'_t} [(\epsilon_{fake}(\tilde{y}'_t, t) - \epsilon_{real}(\tilde{y}'_t, t)) \frac{d\epsilon_{\theta}}{d\theta}] \quad (6)$$

Here, s_{real} and s_{fake} are the score functions learned by multi-step diffusion model ϵ_{real} and ϵ_{fake} respectively. To ensure the fake score network ϵ_{fake} accurately tracks the evolving distribution of the one-step diffusion model, we update it using a standard denoising loss:

$$L_{fake} = \mathbb{E}_{t, \tilde{y}'_t} \|\epsilon_{fake}(\tilde{y}'_t, t) - \tilde{y}_0\|_2^2 \quad (7)$$

The improved version of diffusion distillation proposed in [19] introduces adversarial training in the latent space to further enhance distribution alignment. A discriminator is trained to differentiate between features extracted from real and generated images, using the following objective:

$$L_{Disc} = \mathbb{E}_{t, x} [Disc(\epsilon_{fake}(\tilde{y}'_t, t)) - Disc(\epsilon_{fake}(E_{VAE}(x) + n_t, t))] \quad (8)$$

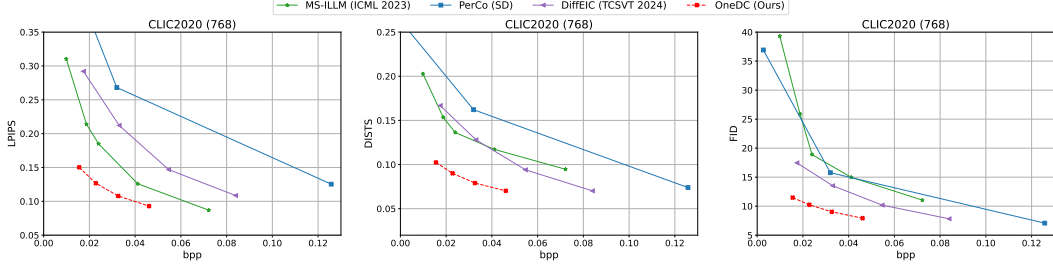


Figure C: Rate–distortion curves on the CLIC2020 at *resize & center-crop*, 256 FID setting, with resized resolution 768.

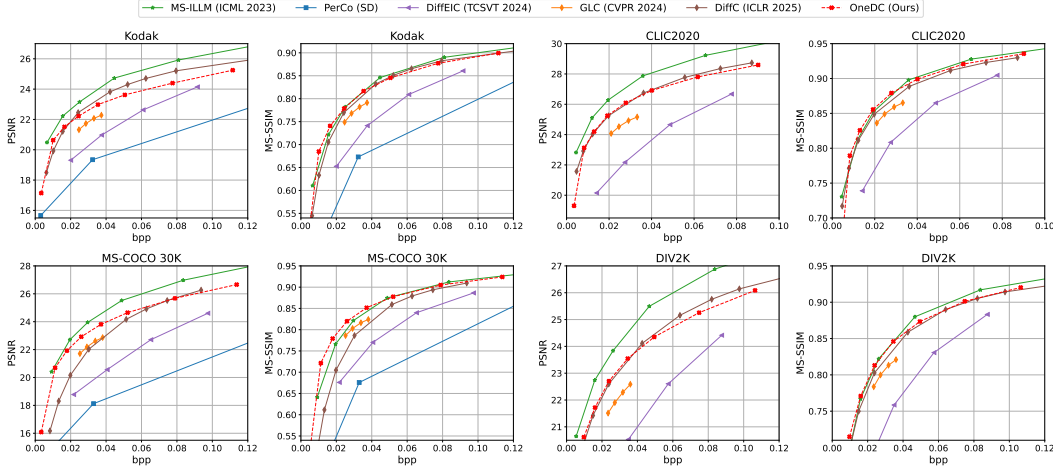


Figure D: Comparison of methods measured by PSNR and MS-SSIM at *full-resolution* setting.

Here, E_{VAE} denotes the encoder of the diffusion model, and n_t is the sampled noise in timestep t . The fake branch is then optimized with an additional adversarial loss:

$$L = L_{\text{fake}} + \sigma L_{\text{Disc}} \quad (9)$$

We follow [19] and set $\sigma = 0.01$. During training, the fake branch and discriminator are each updated 10 times for every update of the one-step generator, ensuring stable adversarial optimization.

Training is conducted on 4×A100 GPUs for 1,000,000 steps. This stage requires around 12 days, as diffusion distillation introduces additional cost in addition to high-resolution training. The learning rate is fixed at 1e-6 (with AdamW) for the one-step generator, fake network, and discriminator. The batch size, cropping strategy and training data are identical to those used in Stage I.

C Model Details

The overall architecture is shown in Fig.E. To ensure better alignment with the latent space of the diffusion model, we extract features from its pretrained VAE encoder. Additionally, inspired by DiffEIC[9], we incorporate embeddings from the original input image to enrich the encoder with complementary spatial and semantic information. To jointly learn compressible latents and capture high-level semantics, we introduce a lightweight U-Net within the analysis transform. Its multi-scale design facilitates effective aggregation of both local textures and global structures. The extracted semantic features are integrated into the hyperprior branch, enhancing its representational capability.

For parameter-efficient adaptation, we insert LoRA [5] layers across all modules of the pretrained one-step diffusion U-Net, setting the LoRA rank to 64. The combined parameter count of the encoder and decoder components ($g_a, h_{\text{enc}}, h_{\text{ctx}}, h_{\text{sem}}$ and g_s) is 394M, while the adapted diffusion U-Net contributes an additional 928M parameters (about 860M for SD1.5 and 68M for LoRA).

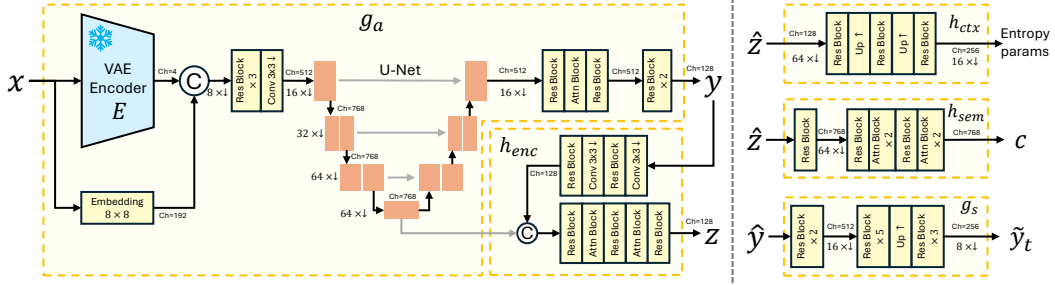


Figure E: Details of our model architecture. The pretrained VAE encoder is from SD 1.5. For the U-Net used in g_a , we use the implementation from the diffusers library [16].

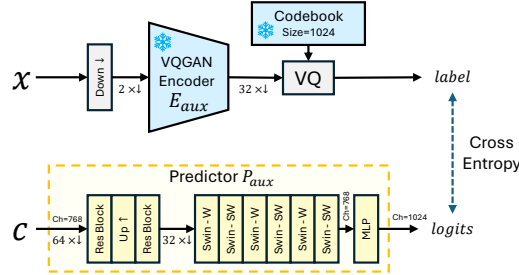


Figure F: Details of the auxiliary code predictor module. For Swin Transformer block [11], we set window size to 16, head dim to 64. W means normal window, SW means shifted window.

Further details of the semantic distillation mechanism are provided in Fig. F. We adopt a Swin Transformer [11] to improve adaptability across different image resolutions during training.

D Social Impact

Positive Aspects. Generative codecs offer substantial benefits by significantly reducing the storage and transmission demands of high-resolution imagery through the synthesis of perceptually convincing content. This improvement in efficiency can help broaden access to high-quality visual media, particularly in bandwidth-limited or resource-constrained environments. The resulting gains in data economy and speed of delivery have promising implications for applications in social communication and personal entertainment.

Negative Aspects. Despite these advantages, generative compression introduces synthesized content that may deviate from the original input, raising concerns about the fidelity and authenticity of reconstructed images. Ongoing research into multi-realism codecs [2] offers a potential path forward, enabling more transparent control over the trade-off between realism and fidelity.

References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. URL <http://www.vision.ee.ethz.ch/~timofter/publications/Agustsson-CVPRW-2017.pdf>.
- [2] Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi-realism image compression with a conditional generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22324–22333, 2023.
- [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.
- [4] Aaron Gokaslan, A Feder Cooper, Jasmine Collins, Landan Seguin, Austin Jacobson, Mihir Patel, Jonathan Frankle, Cory Stephenson, and Volodymyr Kuleshov. Commoncanvas: An open diffusion model trained with creative-commons images. *arXiv preprint arXiv:2310.16825*, 2023.

- [5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [6] 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.
- [7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.
- [8] Nikolai Körber, Eduard Kromer, Andreas Siebert, Sascha Hauke, Daniel Mueller-Gritschneider, and Björn Schuller. Perco (SD): Open perceptual compression. In *Workshop on Machine Learning and Compression, NeurIPS 2024*, 2024. URL <https://openreview.net/forum?id=8xvygfdRWy>.
- [9] 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.
- [10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- [11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [13] 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.
- [14] 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.
- [15] Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models. *arXiv preprint arXiv:2502.01189*, 2025.
- [16] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022.
- [17] Jeremy Vonderfecht and Feng Liu. Lossy compression with pretrained diffusion models. *arXiv preprint arXiv:2501.09815*, 2025.
- [18] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003. doi: 10.1109/ACSSC.2003.1292216.
- [19] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.
- [20] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.

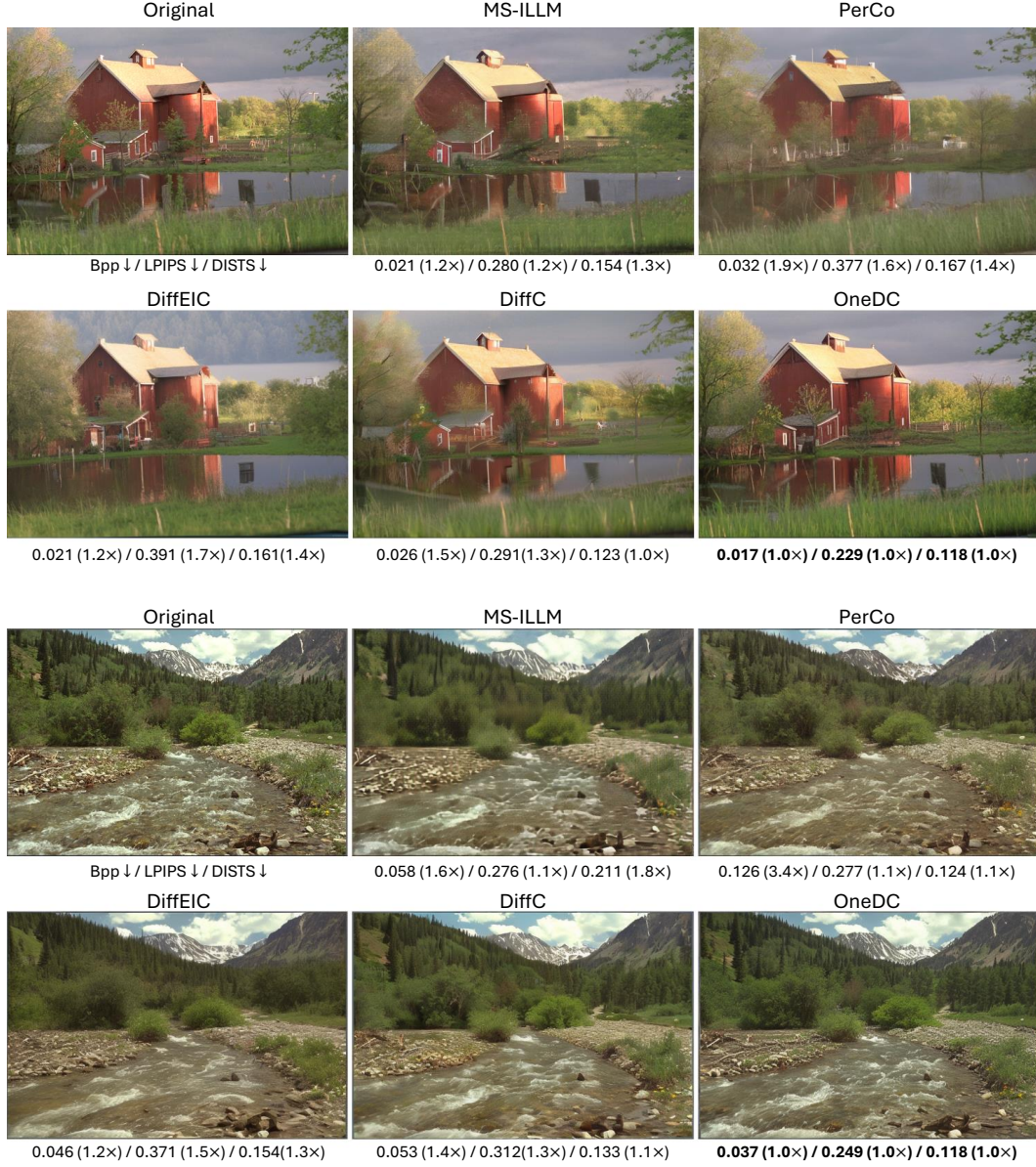


Figure G: Visual results on the Kodak dataset. The VAE-based MS-ILLM exhibits noticeable artifacts. Compared with our OneDC, the previous SOTA multi-step diffusion codec DiffC requires at least 1.4× bitrate on these two examples. Zoom in for better view.

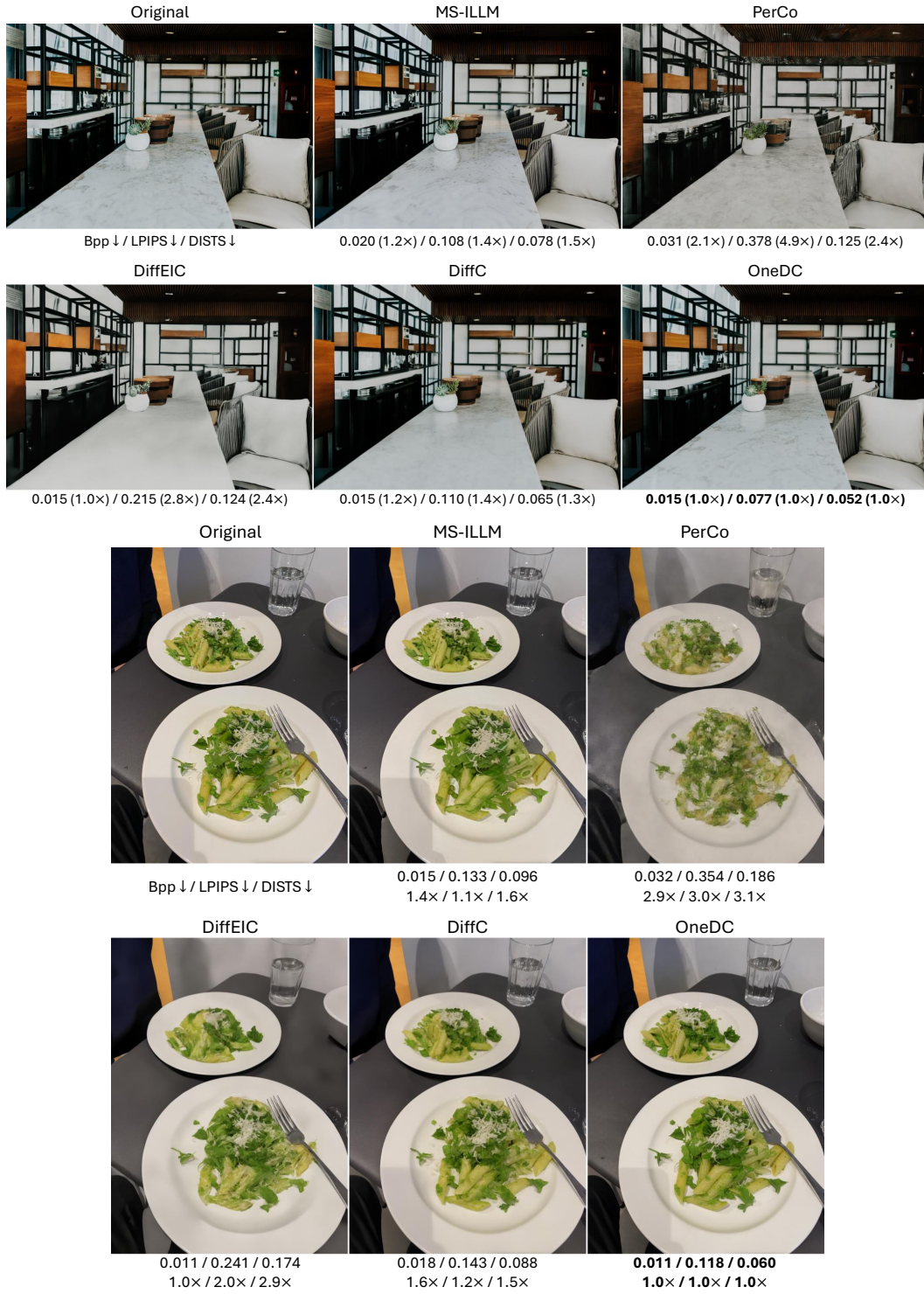


Figure H: Visual results on the CLIC2020 test set. Zoom in for better view.














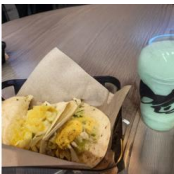


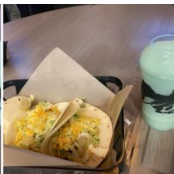
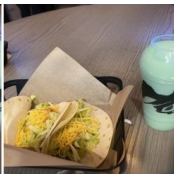
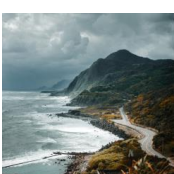
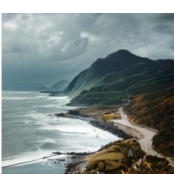

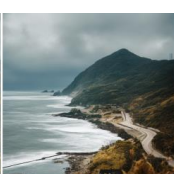
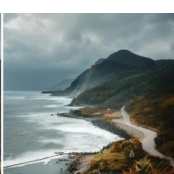
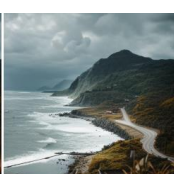






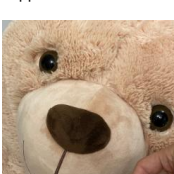
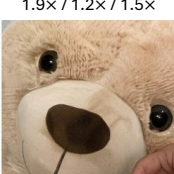
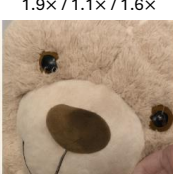
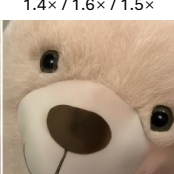
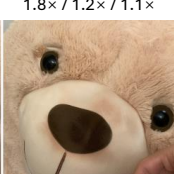
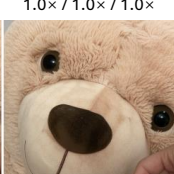
Original	MS-ILLM	PerCo	DiffEIC	DiffC	OneDC
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.026 / 0.236 / 0.162 1.4× / 1.2× / 1.5×	0.032 / 0.345 / 0.188 1.7× / 1.8× / 1.7×	0.022 / 0.370 / 0.206 1.2× / 1.9× / 1.9×	0.029 / 0.248 / 0.125 1.5× / 1.3× / 1.1×	0.019 / 0.195 / 0.109 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.021 / 0.250 / 0.137 1.2× / 1.2× / 1.4×	0.032 / 0.353 / 0.208 1.9× / 1.7× / 2.1×	0.017 / 0.365 / 0.198 1.0× / 1.8× / 2.0×	0.026 / 0.264 / 0.118 1.5× / 1.3× / 1.2×	0.017 / 0.202 / 0.097 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.021 / 0.208 / 0.119 1.4× / 1.2× / 1.4×	0.032 / 0.305 / 0.166 2.1× / 1.8× / 1.9×	0.017 / 0.321 / 0.163 1.1× / 1.9× / 1.9×	0.023 / 0.227 / 0.089 1.5× / 1.4× / 1.0×	0.015 / 0.167 / 0.086 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.022 / 0.223 / 0.160 1.6× / 1.2× / 1.4×	0.032 / 0.337 / 0.213 2.3× / 1.8× / 1.8×	0.019 / 0.335 / 0.175 1.4× / 1.8× / 1.5×	0.025 / 0.260 / 0.116 1.8× / 1.4× / 1.0×	0.014 / 0.185 / 0.116 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.032 / 0.264 / 0.124 1.9× / 1.2× / 1.5×	0.032 / 0.344 / 0.134 1.9× / 1.1× / 1.6×	0.024 / 0.360 / 0.124 1.4× / 1.6× / 1.5×	0.030 / 0.269 / 0.087 1.8× / 1.2× / 1.1×	0.017 / 0.223 / 0.082 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.016 / 0.222 / 0.122 1.3× / 1.2× / 1.2×	0.032 / 0.283 / 0.169 2.7× / 1.6× / 1.7×	0.015 / 0.363 / 0.221 1.3× / 2.0× / 2.2×	0.023 / 0.243 / 0.124 1.9× / 1.4× / 1.2×	0.012 / 0.180 / 0.100 1.0× / 1.0× / 1.0×

Figure I: Visual results on the CLIC2020 test set (768×768). Compared with OneDC, the previous SOTA multi-step diffusion method DiffC requires ≥ 1.5 higher bitrate while still producing slightly lower reconstruction quality on these examples. Zoom in for better view.





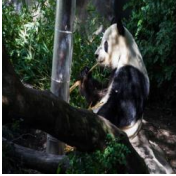



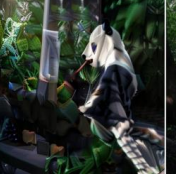







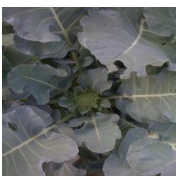

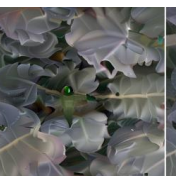
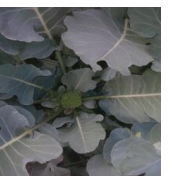




Original	MS-ILLM	PerCo	DiffEIC	DiffC	OneDC
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.030 / 0.246 / 0.123 1.4× / 1.4× / 1.2×	0.033 / 0.376 / 0.149 1.6× / 2.2× / 1.4×	0.024 / 0.369 / 0.155 1.1× / 2.1× / 1.5×	0.023 / 0.386 / 0.181 1.1× / 2.2× / 1.8×	0.021 / 0.174 / 0.103 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.032 / 0.248 / 0.146 1.7× / 1.1× / 1.1×	0.033 / 0.428 / 0.210 1.7× / 1.9× / 1.6×	0.027 / 0.377 / 0.165 1.4× / 1.7× / 1.3×	0.025 / 0.448 / 0.271 1.3× / 2.0× / 2.1×	0.019 / 0.220 / 0.130 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.051 / 0.229 / 0.147 2.0× / 1.1× / 1.3×	0.033 / 0.393 / 0.194 1.3× / 1.9× / 1.7×	0.032 / 0.346 / 0.140 1.2× / 1.7× / 1.3×	0.027 / 0.351 / 0.165 1.0× / 1.7× / 1.5×	0.026 / 0.206 / 0.111 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.042 / 0.250 / 0.170 1.8× / 1.2× / 1.5×	0.033 / 0.403 / 0.186 1.4× / 1.9× / 1.6×	0.029 / 0.375 / 0.175 1.2× / 1.8× / 1.5×	0.026 / 0.365 / 0.178 1.1× / 1.7× / 1.5×	0.024 / 0.209 / 0.117 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.023 / 0.299 / 0.220 1.0× / 2.9× / 1.7×	0.033 / 0.400 / 0.212 1.5× / 3.8× / 1.7×	0.024 / 0.421 / 0.209 1.1× / 4.0× / 1.7×	0.023 / 0.434 / 0.293 1.0× / 4.2× / 2.3×	0.022 / 0.104 / 0.126 1.0× / 1.0× / 1.0×
					
Bpp ↓ / LPIPS ↓ / DISTs ↓	0.039 / 0.231 / 0.147 1.9× / 2.7× / 1.6×	0.033 / 0.347 / 0.147 1.6× / 4.0× / 1.6×	0.025 / 0.344 / 0.187 1.2× / 4.0× / 2.0×	0.022 / 0.360 / 0.222 1.0× / 4.1× / 2.4×	0.021 / 0.087 / 0.094 1.0× / 1.0× / 1.0×

Figure J: Visual results on the MS-COCO 30K dataset. PerCo, the previous best-performing method in terms of FID, requires ≥ 1.4 bitrate over OneDC, but still results in suboptimal fidelity. Zoom in for better view.

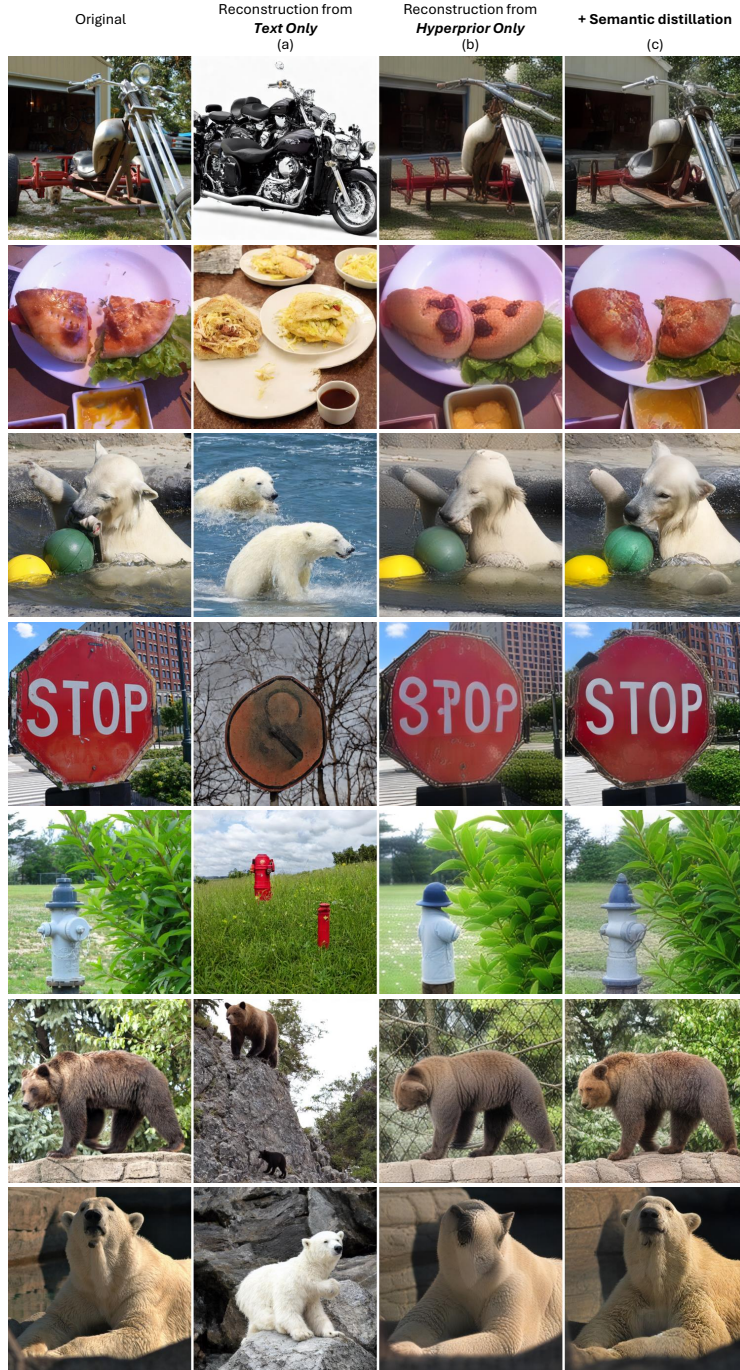


Figure K: Reconstructions from different semantic guidance. (a) Text prompts struggle to capture complex visual semantics, resulting in severe distortions when using a pretrained text-to-image one-step diffusion model [19]. (b) We finetune the model [19] for hyperprior-to-image generation. Hyperprior guidance yields more faithful reconstructions. (c) Our proposed semantic distillation further improves object-level accuracy.

Table C: Compression quantitative evaluations, for the Kodak dataset with full-resolution.

Method	BPP	PSNR	MS-SSIM	LPIPS	DISTS
MS-ILLM	0.0066	20.478	0.611	0.438	0.280
	0.0156	22.224	0.722	0.292	0.177
	0.0250	23.155	0.782	0.228	0.147
	0.0447	24.731	0.846	0.158	0.135
	0.0809	25.922	0.890	0.110	0.109
	0.1535	27.532	0.928	0.073	0.081
	0.2962	29.634	0.960	0.045	0.061
PerCo (SD)	0.0031	15.662	0.418	0.529	0.234
	0.0324	19.344	0.673	0.299	0.162
	0.1261	22.967	0.847	0.141	0.080
DiffEIC	0.0200	19.308	0.653	0.326	0.169
	0.0375	20.970	0.741	0.242	0.134
	0.0610	22.631	0.809	0.173	0.098
	0.0916	24.147	0.861	0.129	0.077
GLC	0.0247	21.320	0.749	0.196	0.113
	0.0286	21.729	0.768	0.180	0.104
	0.0331	22.071	0.782	0.168	0.098
	0.0374	22.279	0.791	0.161	0.095
DiffC	0.0063	18.494	0.545	0.450	0.201
	0.0102	19.926	0.633	0.363	0.165
	0.0155	21.212	0.705	0.288	0.139
	0.0242	22.453	0.770	0.225	0.112
	0.0423	23.823	0.832	0.163	0.087
	0.0522	24.304	0.851	0.144	0.080
	0.0625	24.704	0.865	0.130	0.074
	0.0795	25.207	0.883	0.113	0.065
	0.1227	25.949	0.905	0.091	0.055
OneDC	0.0034	17.141	0.472	0.380	0.204
	0.0101	20.631	0.685	0.220	0.131
	0.0165	21.513	0.741	0.183	0.110
	0.0245	22.220	0.779	0.154	0.097
	0.0354	22.979	0.817	0.133	0.085
	0.0506	23.620	0.845	0.114	0.076
	0.0775	24.396	0.877	0.096	0.066
	0.1115	25.248	0.899	0.083	0.057

Table D: Compression quantitative evaluations, for the CLIC2020 test set with full-resolution.

Method	BPP	PSNR	MS-SSIM	LPIPS	DISTS	FID
MS-ILLM	0.0045	22.815	0.731	0.365	0.216	44.954
	0.0120	25.100	0.813	0.225	0.131	15.329
	0.0195	26.275	0.853	0.173	0.106	8.748
	0.0359	27.875	0.898	0.114	0.086	6.261
	0.0654	29.229	0.928	0.080	0.066	4.481
	0.1240	30.865	0.953	0.054	0.048	2.645
	0.2377	32.833	0.972	0.035	0.035	1.660
PerCo (SD)	0.0022	15.340	0.473	0.589	0.326	76.658
	0.0315	16.733	0.537	0.389	0.194	18.039
	0.1249	17.419	0.554	0.306	0.091	5.013
DiffEIC	0.0142	20.152	0.739	0.308	0.185	18.427
	0.0275	22.173	0.808	0.222	0.132	11.039
	0.0487	24.656	0.865	0.153	0.089	7.278
	0.0776	26.673	0.905	0.109	0.060	4.942
DiffC	0.0047	21.571	0.717	0.311	0.179	20.076
	0.0079	22.914	0.771	0.253	0.144	12.332
	0.0123	24.040	0.811	0.210	0.117	8.957
	0.0197	25.241	0.849	0.171	0.092	6.813
	0.0362	26.745	0.889	0.128	0.067	5.252
	0.0556	27.772	0.911	0.103	0.053	4.414
	0.0724	28.357	0.923	0.090	0.046	3.929
	0.0872	28.733	0.930	0.082	0.042	3.685
OneDC	0.0035	19.305	0.629	0.290	0.169	14.885
	0.0083	23.129	0.789	0.164	0.089	6.223
	0.0130	24.201	0.826	0.139	0.077	5.560
	0.0192	25.248	0.856	0.119	0.068	4.979
	0.0279	26.096	0.879	0.103	0.060	4.234
	0.0401	26.912	0.899	0.090	0.053	3.581
	0.0617	27.804	0.921	0.076	0.044	2.977
	0.0902	28.596	0.936	0.065	0.037	2.410

Table E: Compression quantitative evaluations, for the MS-COCO 30K dataset at 512×512 resolution.

Method	BPP	PSNR	MS-SSIM	LPIPS	DISTS	FID
MS-ILLM	0.0092	20.411	0.641	0.397	0.255	72.693
	0.0196	22.708	0.766	0.257	0.169	17.992
	0.0296	23.948	0.821	0.200	0.145	9.041
	0.0488	25.517	0.875	0.138	0.122	4.100
	0.0835	26.972	0.912	0.095	0.099	2.032
	0.1510	28.734	0.944	0.063	0.075	0.990
	0.2850	30.930	0.969	0.039	0.058	0.457
PerCo (SD)	0.0036	14.126	0.388	0.545	0.245	4.467
	0.0329	18.124	0.676	0.311	0.159	2.748
	0.1267	22.802	0.869	0.134	0.080	1.152
DiffEIC	0.0217	18.778	0.676	0.318	0.171	6.151
	0.0407	20.564	0.770	0.229	0.133	3.929
	0.0653	22.710	0.840	0.159	0.099	2.578
	0.0975	24.606	0.886	0.116	0.077	1.911
DiffC	0.0083	16.170	0.495	0.524	0.271	90.989
	0.0133	18.306	0.611	0.427	0.229	57.542
	0.0198	20.159	0.705	0.334	0.187	28.909
	0.0303	22.029	0.786	0.241	0.140	9.763
	0.0514	24.170	0.859	0.155	0.095	2.805
	0.0628	24.915	0.879	0.132	0.082	2.135
	0.0745	25.520	0.894	0.115	0.074	1.809
	0.094	26.264	0.910	0.098	0.064	1.518
OneDC	0.0034	16.083	0.460	0.376	0.211	13.607
	0.0112	20.696	0.721	0.201	0.128	3.496
	0.0179	21.924	0.779	0.166	0.111	2.817
	0.0260	22.924	0.820	0.139	0.097	2.379
	0.0371	23.809	0.852	0.118	0.086	2.044
	0.0521	24.650	0.878	0.101	0.076	1.719
	0.0789	25.674	0.905	0.082	0.064	1.365
	0.1137	26.662	0.924	0.069	0.055	1.043

Table F: Compression quantitative evaluations, for the DIV2K test set at full-resolution.

Method	BPP	PSNR	MS-SSIM	LPIPS	DISTS	FID
MS-ILLM	0.0054	20.647	0.647	0.396	0.234	79.455
	0.0159	22.741	0.767	0.254	0.143	33.303
	0.0262	23.837	0.822	0.198	0.116	22.428
	0.0467	25.493	0.880	0.138	0.102	18.537
	0.0837	26.878	0.917	0.097	0.080	13.574
	0.1547	28.503	0.946	0.065	0.055	8.070
	0.2895	30.545	0.969	0.040	0.040	5.123
PerCo (SD)	0.0022	14.529	0.381	0.570	0.320	95.345
	0.0316	15.411	0.432	0.417	0.218	36.671
	0.1249	15.340	0.405	0.355	0.105	11.314
DiffEIC	0.0182	18.615	0.663	0.327	0.190	34.527
	0.0349	20.532	0.759	0.241	0.132	21.656
	0.0575	22.604	0.831	0.172	0.089	14.026
	0.0876	24.415	0.883	0.125	0.061	9.644
DiffC	0.0056	19.091	0.623	0.353	0.198	45.767
	0.0096	20.345	0.695	0.291	0.158	32.086
	0.0149	21.422	0.750	0.244	0.127	24.270
	0.0238	22.599	0.803	0.199	0.098	18.541
	0.0427	24.113	0.859	0.149	0.071	14.098
	0.0640	25.158	0.890	0.120	0.056	11.859
	0.0820	25.755	0.905	0.104	0.048	10.622
	0.0976	26.143	0.914	0.096	0.044	9.938
	0.1273	26.645	0.925	0.085	0.039	8.983
OneDC	0.0097	20.615	0.715	0.200	0.103	15.874
	0.0161	21.719	0.771	0.168	0.086	13.680
	0.0239	22.703	0.813	0.144	0.077	12.365
	0.0345	23.546	0.846	0.124	0.067	11.242
	0.0495	24.351	0.873	0.108	0.060	10.096
	0.0749	25.257	0.901	0.090	0.050	8.325
	0.1064	26.086	0.920	0.077	0.040	6.822

Table G: Compression quantitative evaluations, for the Kodak dataset at *resize & center-crop* setting with 512×512 resolution.

Method	BPP	PSNR	LPIPS	FID
MS-ILLM	0.085	25.683	0.11	36.08
	0.159	27.296	0.072	28.556
	0.304	29.395	0.044	24.448
PerCo (SD)	0.033	19.017	0.307	37.019
	0.127	22.325	0.145	26.418
DDCM	0.03	22.066	0.222	32.031
	0.038	22.551	0.19	29.117
	0.05	23.013	0.161	25.647
	0.095	23.606	0.138	24.215
	0.149	24.069	0.124	23.199
OneDC	0.0109	20.259	0.223	36.862
	0.0177	21.219	0.183	33.075
	0.0257	21.970	0.154	30.947
	0.0369	22.718	0.132	29.103
	0.0524	23.363	0.113	27.428
	0.0795	24.202	0.095	25.452
	0.1139	25.021	0.082	23.448

Table H: Compression quantitative evaluations, for the CLIC2020 test set at *resize & center-crop* setting with 768×768 resolution.

Method	BPP	PSNR	LPIPS	FID
MS-ILLM	0.006	21.304	0.447	62.327
	0.009	22.703	0.34	41.479
	0.072	27.808	0.084	7.852
	0.134	29.567	0.055	5.966
PerCo (SD)	0.003	15.339	0.517	30.409
	0.032	19.018	0.269	12.869
	0.126	23.387	0.122	5.419
DDCM	0.007	19.672	0.404	23.862
	0.008	20.532	0.354	19.521
	0.01	21.207	0.314	15.559
	0.014	22.116	0.262	11.362
	0.017	22.722	0.227	8.722
	0.022	23.366	0.192	6.753
	0.042	24.136	0.156	5.051
	0.066	24.739	0.133	4.549
	0.137	25.65	0.108	4.132
OneDC	0.0034	17.619	0.326	15.325
	0.0098	22.378	0.172	7.953
	0.0155	23.600	0.140	7.390
	0.0228	24.641	0.117	7.011
	0.0326	25.532	0.099	6.538
	0.0461	26.416	0.084	5.874
	0.0694	27.463	0.069	4.915
	0.0993	28.471	0.058	4.046