One-Step Diffusion-Based Image Compression with Semantic Distillation

Naifu Xue[1*], **Zhaoyang Jia**[2*], **Jiahao Li**[3], **Bin Li**[3], **Yuan Zhang**[1], **Yan Lu**[3]
[1] Communication University of China [2] University of Science and Technology of China
[3] Microsoft Research Asia
`{aaronxuenf, yzhang}@cuc.edu.cn, {jzy_ustc}@mail.ustc.edu.cn`
`{li.jiahao, libin, yanlu}@microsoft.com`

Abstract

While recent diffusion-based generative image codecs have shown impressive performance, their iterative sampling process introduces unpleasant latency. In this work, we revisit the design of a diffusion-based codec and argue that multi-step sampling is not necessary for generative compression. Based on this insight, we propose OneDC, a **One**-step **D**iffusion-based generative image **C**odec—that integrates a latent compression module with a one-step diffusion generator. Recognizing the critical role of semantic guidance in one-step diffusion, we propose using the hyperprior as a semantic signal, overcoming the limitations of text prompts in representing complex visual content. To further enhance the semantic capability of the hyperprior, we introduce a semantic distillation mechanism that transfers knowledge from a pretrained generative tokenizer to the hyperprior codec. Additionally, we adopt a hybrid pixel- and latent-domain optimization to jointly enhance both reconstruction fidelity and perceptual realism. Extensive experiments demonstrate that OneDC achieves SOTA perceptual quality even with one-step generation, offering over **39%** bitrate reduction and **20**× faster decoding compared to prior multi-step diffusion-based codecs. Project: `https://onedc-codec.github.io/`

1 Introduction

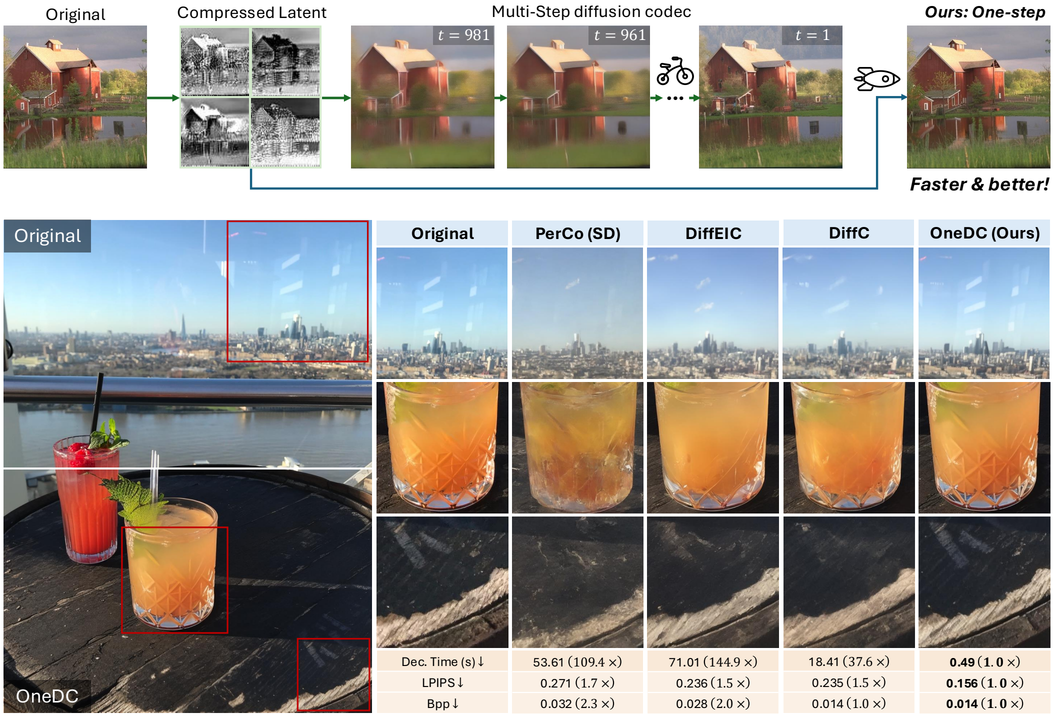
The rapid growth in image data has led to increased storage and transmission costs, heightening the need for efficient, high-quality image compression techniques. While VAE-based learned image compression (LIC) [36, 71] has surpassed traditional codecs (e.g., VVC [8]) in rate-distortion (RD) performance, it often produces blurry details at low bitrates [41] since they are typically optimized for objective distortion rather than human perception [7]. To mitigate this, some approaches [41, 44] focus on optimizing visual quality by incorporating perceptual metrics and adversarial losses to enable generative image compression. Despite these advances, such generative models tend to introduce artifacts under extreme compression, degrading the realism of reconstructed images.

Recently, diffusion-based generative codecs [9, 33] have been introduced to enhance reconstruction quality by leveraging the powerful content synthesis capabilities of pretrained models. While these methods significantly improve perceptual realism, they may occasionally generate content that deviates from the original input, thereby compromising reconstruction fidelity. In addition, their inherently iterative sampling process leads to substantial computational overhead, making them notably slower than conventional VAE-based codecs.

In standard diffusion-based image generation, the model progressively refines a noisy signal through iterative denoising, beginning with coarse structures and gradually synthesizing high-frequency details [65]. However, when applied to image compression, the task changes: at low bitrates, codecs typically

[*] Naifu Xue and Zhaoyang Jia are visiting students at Microsoft Research Asia.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



	PerCo (SD)	DiffEIC	DiffC	OneDC (Ours)
Dec. Time (s)↓	53.61 (109.4 ×)	71.01 (144.9 ×)	18.41 (37.6 ×)	**0.49** (**1.0** ×)
LPIPS↓	0.271 (1.7 ×)	0.236 (1.5 ×)	0.235 (1.5 ×)	**0.156** (**1.0** ×)
Bpp↓	0.032 (2.3 ×)	0.028 (2.0 ×)	0.014 (1.0 ×)	**0.014** (**1.0** ×)

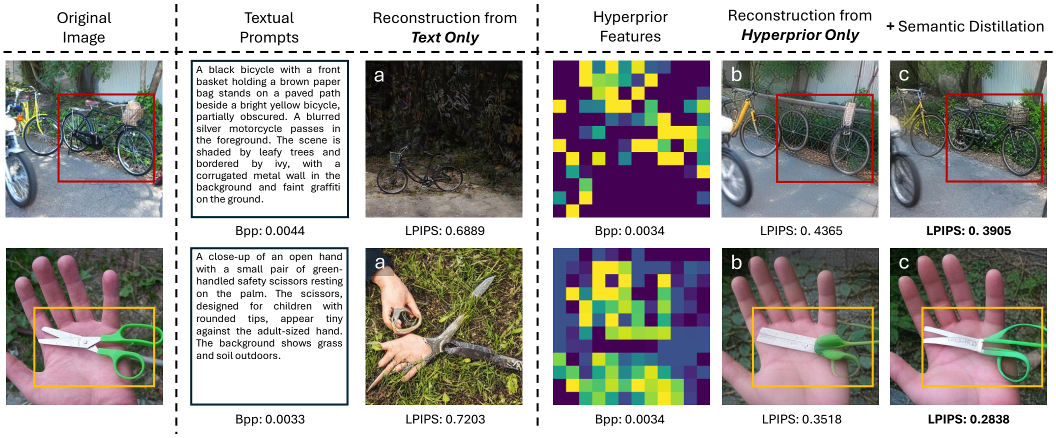
Figure 1: *Top*: multi-step sampling is not essential for image compression; intermediate results are from DiffEIC [33]. *Bottom*: Visual comparisons including existing open-sourced multi-step diffusion codecs [9, 33, 57] and our proposed **one-step** codec. Our method achieves the highest visual quality at the lowest bitrate while offering significantly faster decoding.

retain coarse structures in the compressed latents. As a result, the decoder is primarily responsible for plausibly reconstructing high-frequency content from the preserved low-frequency information. This observation motivates our central hypothesis: *given the compressed latent, multi-step sampling is not necessary for decoding, and a carefully designed one-step alternative could suffice.* Although recent advances in one-step diffusion [53, 66, 25] offer promising tools for accelerating inference, their potential remains underexplored in the context of image compression.

In this paper, we introduce **OneDC** (**One**-step **D**iffusion-based generative **C**odec), a novel framework for ultra-low bitrate image coding. OneDC integrates a latent compression module with a one-step diffusion generator: the former encodes the image into compact latents, while the latter synthesizes high-frequency details conditioned on the latent. Since the pixel-domain training is insufficient for guaranteeing perceptual quality at low bitrates [22], we adopt a hybrid-domain training strategy. Specifically, we combine a pixel-domain perceptual loss to promote fidelity with a latent-domain diffusion distillation objective [66] to enhance realism. This design allows OneDC to effectively leverage the pretrained diffusion model while achieving a balanced trade-off among compression ratio, perceptual quality, and decoding efficiency, as illustrated in Fig. 1.

Moreover, we further explore the role of semantic guidance (i.e., the input of the cross-attention layers [50]) within our framework. In one-step diffusion, such guidance is essential to compensate for the absence of multi-step refinement (see ablation in Section 4.3). While existing diffusion models have utilized text prompts for conditioning, we argue that text is suboptimal for image compression due to two limitations: (1) natural language struggles to capture fine-grained or localized visual semantics, and (2) generating high-quality captions typically requires large-scale vision-language models (e.g., the large BLIP2 [32] used in [9]), introducing substantial computational overhead. This raises a central question: *Can we design more effective semantic representations than text for guiding one-step diffusion-based codecs—without incurring excessive computational cost?*

Recent studies have shown that the hyperprior in VAE-based codecs can capture high-level semantic information [22, 48]. Compared to textual prompts, hyperpriors provide more precise and spatially aligned semantic cues, making them well-suited for representing localized semantics in high-resolution images (see Fig.2). This insight motivates us to further enhance the representational



A black bicycle with a front basket holding a brown paper bag stands on a paved path beside a bright yellow bicycle, partially obscured. A blurred silver motorcycle passes in the foreground. The scene is shaded by leafy trees and bordered by ivy, with a corrugated metal wall in the background and faint graffiti on the ground.

A close-up of an open hand with a small pair of green-handled safety scissors resting on the palm. The scissors, designed for children with rounded tips, appear tiny against the adult-sized hand. The background shows grass and soil outdoors.

Figure 2: Reconstructions from different semantic guidance. (a) Text prompts (from GPT-4o [47]) struggle to capture complex visual semantics, resulting in severe distortions when using a pretrained text-to-image one-step diffusion model [66]. (b) We finetune the model [66] for hyperprior-to-image generation. Hyperprior guidance yields more faithful reconstructions. (c) Our proposed semantic distillation further improves object-level accuracy, particularly in the highlighted regions.

capability of the hyperprior. Inspired by generative tokenizers [15], where codebooks encode rich semantic content, we propose hyperprior semantic distillation—a training strategy that transfers semantic knowledge from a pretrained tokenizer to the hyperprior through a code prediction module [72]. As illustrated in Fig.2, this approach results in reconstructions that are more semantically accurate and visually coherent.

Experiments show that OneDC achieves state-of-the-art (SOTA) performance in generative image compression, delivering over **39%** bitrate reduction and **20**× faster decoding compared to existing multi-step diffusion codecs. Ablation studies further validate the effectiveness of our optimization strategy and the semantic distillation method for hyperprior. Our contributions are as follows:

- We propose OneDC, a one-step diffusion codec comprising a latent compression module for compact feature encoding and a one-step diffusion generator for fast, high-quality decoding. A hybrid-domain training strategy further enhances both fidelity and perceptual realism.
- We identify the importance of high-level semantic guidance in one-step diffusion and highlight the limitations of text embeddings. To improve reconstruction quality, we introduce hyperprior features as an alternative and enhance them through semantic distillation.
- Extensive experiments show that OneDC achieves SOTA compression performance while offering significantly faster decoding than existing diffusion-based codecs, demonstrating the potential of one-step diffusion in generative compression.

2 Related Work

VAE-based Learned Image Compression. Learned image compression has made rapid progress in RD performance. Ballé et al. [4] introduced an end-to-end framework with analysis/synthesis transforms and a factorized entropy model. The hyperprior model [5] enhanced entropy modeling by encoding side information (hyperprior) to predict distribution parameters, substantially improving RD performance. Subsequent work combined hyperpriors with spatial context [43] and adopts more expressive distribution models such as Gaussian mixtures [11]. Recent studies further refined transform architectures for stronger nonlinear representation [36] and developed advanced entropy models for more compact symbol coding [23], together surpassing the VVC-Intra baseline [8]. Nevertheless, these models are typically optimized with distortion-oriented objectives such as MSE or MS-SSIM [60], which often over-smooth textures and degrade perceptual quality at low bitrates [41], even with SOTA transforms and entropy models [36, 71, 58].

VAE-based Generative Image Compression. To improve perceptual quality at low bitrates, GAN-based approaches such as Agustsson et al. [2] and HiFiC [41] train VAE codecs to synthesize high-frequency details using perceptual losses [24] and adversarial discriminators. MS-ILLM [44]

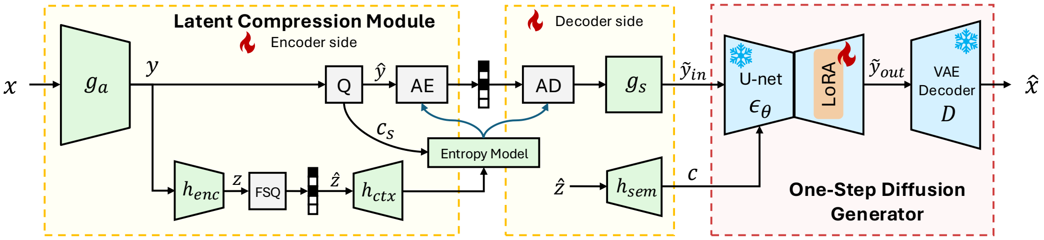


Figure 3: Overview of the OneDC framework. Q denotes scalar quantization, and FSQ stands for finite scalar quantization. AE and AD refer to the arithmetic encoder and decoder, respectively. h_{ctx} and h_{sem} represent the context and semantic decoders used in the hyperprior branch.

improves realism through a local likelihood model, while EGIC [27] introduces semantic-aware feedback via segmentation-conditioned discriminators. Other methods enhance compression by integrating generative tokenizers: Mao et al. [40] utilize VQGAN [15] for compression, Xue et al. [63] unify token generation and entropy coding, and Jia et al. [22] propose GLC that improves token compression via transform coding. More recently, dual-branch codecs [39, 64] have been introduced to support more compact and flexible latent representations. While these methods highlight the strong feature learning capacity of VAE-based codecs, their reconstruction quality at extremely low bitrates remains limited by model capacity and training scale.

Diffusion-based Generative Image Compression. Diffusion models have rapidly evolved in recent years, progressing from theoretical foundations [51, 20] to practical high-quality implementations [50]. Building on these advances, recent studies [29, 9, 33, 57, 62] have explored the use of generative priors from large-scale pretrained diffusion models in image compression, demonstrating improved perceptual realism compared to conventional VAE-based approaches. Eric et al. [29] combine pretrained diffusion with ControlNet [69] to reconstruct images from captions and compressed sketches. PerCo [9] fine-tunes diffusion models using captions and quantized features, with captions generated by a large BLIP2 model [32]. DiffEIC [33] conditions diffusion models on VAE latents, showing strong performance even without text, and their subsequent work [34] improves encoder and introduces relay fine-tuning. DiffC [57] demonstrates zero-shot compression with diffusion models using reverse-channel coding. DDCM [46] replaces the continuous Gaussian noise in sampling with discrete codebook selections for compression task. While these approaches improve realism at ultra-low bitrates, they often struggle to maintain high fidelity. Moreover, multi-step sampling in DDIM [52] introduces substantial decoding overhead, limiting their practicality. By contrast, our OneDC achieves high fidelity and realism with significantly faster decoding via one-step sampling. While prior work [33] suggests high-level semantics (e.g., text) may be optional, we find such guidance remains crucial in our one-step diffusion for image decoding task, as detailed in Section 4.3.

One-step Diffusion Models. To reduce the computational cost, recent efforts have focused on distilling multi-step models into efficient one-step generators. Song et al. [53] introduce Consistency Models, which achieve few-step generation by enforcing self-consistency along diffusion trajectories. Yin et al. [67] propose Distribution Matching Distillation (DMD), which minimizes an approximate KL divergence between real and generated image distributions modeled by multi-step teacher networks. Their subsequent work, DMD2 [66], further improves generation quality by introducing a diffusion-GAN framework and enhanced training strategies. Based on DMD, Song et al. [54] introduce class-specific student models to improve performance across diverse categories. SwiftBrush series [45, 12] adopt a LoRA adaptation [21] for variational score distillation and further optimize diversity by integrating a clamped CLIP loss. Independently, Kang et al. [25] employ perceptual loss supervision and a multi-scale U-Net discriminator for one-step distillation. These models have also been extended to low-level vision tasks such as super-resolution [14, 61, 59, 26] and image restoration [18], highlighting the potential of one-step diffusion beyond generation. Inspired by these advances, we explore its application in generative image compression, a setting that remains underexplored.

3 Methodology

We propose **OneDC**, a one-step diffusion-based generative image compression framework that achieves high-quality reconstruction with low-latency decoding under ultra-low bitrate constraints. Section 3.1 introduces the coding pipeline of our method, consisting of a latent compression module

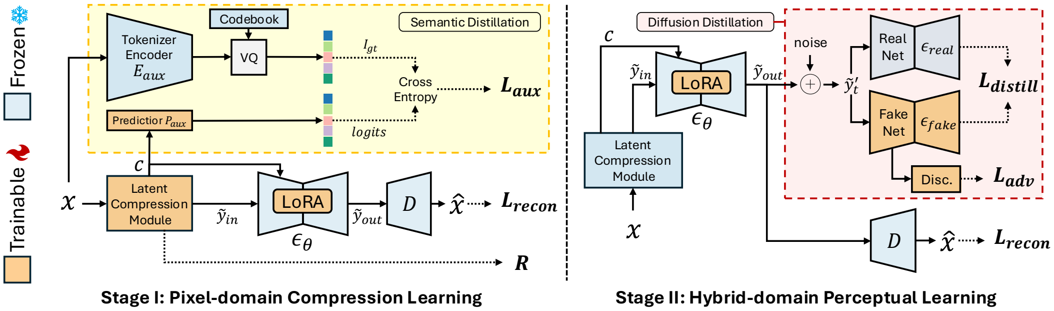


Figure 4: Two stage training pipeline of OneDC. The codebook in semantic distillation is initialized from the pretrained tokenizer, and the discriminator in diffusion distillation is abbreviated as Disc.

that encodes the input image into a compact latent representation and a one-step diffusion generator that synthesizes reconstructions with realistic details. Given the critical role of semantic guidance in our one-step diffusion and the limitations of textual prompts in compression scenarios, Section 3.2 introduces the use of the hyperprior as an alternative conditioning signal. To further tap the potential of the hyperprior, we propose a semantic distillation strategy that transfers knowledge from a pretrained generative tokenizer to the hyperprior codec, thereby improving semantic accuracy. Finally, Section 3.3 presents our training scheme, which combines pixel-domain and latent-domain objectives to jointly optimize reconstruction fidelity and perceptual realism.

3.1 Framework Overview

Fig. 3 shows the overview of our OneDC framework. On the encoder side, an analysis transform encodes the input x into a compact latent representation $y = g_a(x)$, which is then quantized to $\hat{y}$. A hyper encoder further processes the latent into a hyperprior $z = h_{enc}(y)$, quantized as $\hat{z}$. Then, the entropy model takes decoded hyperprior context $c_h = h_{ctx}(\hat{z})$ and spatial context c_s to predict the distribution of $\hat{y}$, enabling bitrate estimation during training and entropy coding during inference.

On the decoder side, the received $\hat{y}$ is passed through a synthesis transform to produce an initial latent $\tilde{y}_{in} = g_s(\hat{y})$ for generation. Given the distinct roles of entropy modeling and semantic processing, where the former estimates low-dimensional distribution parameters and the latter represents high-dimensional visual contents, we introduce an additional semantic decoder for the hyperprior to better address their different requirements. It extracts semantic guidance $c = h_{sem}(\hat{z})$ from the quantized hyperprior, which is injected into the cross-attention layers of the one-step diffusion generator. The diffusion generator refines the initial latent in only one step: $\tilde{y}_{out} = \epsilon_\theta(\tilde{y}_{in}, c)$, which is then decoded by a pretrained VAE decoder to produce the final reconstruction $\hat{x} = D(\tilde{y}_{out})$.

3.2 Semantic Guidance with Hyperprior

3.2.1 From Text to Hyperprior

In the one-step diffusion setting, semantic guidance (the input of the cross-attention layers [50]) plays an even more critical role than in conventional multi-step diffusion. Unlike multi-step methods, which iteratively refine the output and can gradually correct semantic inconsistencies, one-step diffusion models rely entirely on a single forward pass, making accurate semantic conditioning essential. While existing diffusion models typically rely on textual prompts to provide high-level semantic guidance, such prompts are suboptimal in the context of natural image compression. Text struggles to capture localized semantics and often fails to describe fine-grained visual attributes—such as object boundaries and textures—particularly in high-resolution imagery. In addition, generating text descriptions typically requires large models [32], introducing significant computational overhead.

Recent work [22] suggests that the categorical distributed hyperprior with vector quantization (VQ) can capture semantic information. Building on this insight, OneDC replaces textual prompts with the categorical hyperprior $\hat{z}$ as semantic guidance for the diffusion model. To compress the hyperprior, we adopt finite scalar quantization (FSQ) [42], a simpler yet more effective alternative to VQ [15], to learn a categorical $\hat{z}$ with semantic information. The hyperprior z is compressed into 7 channels

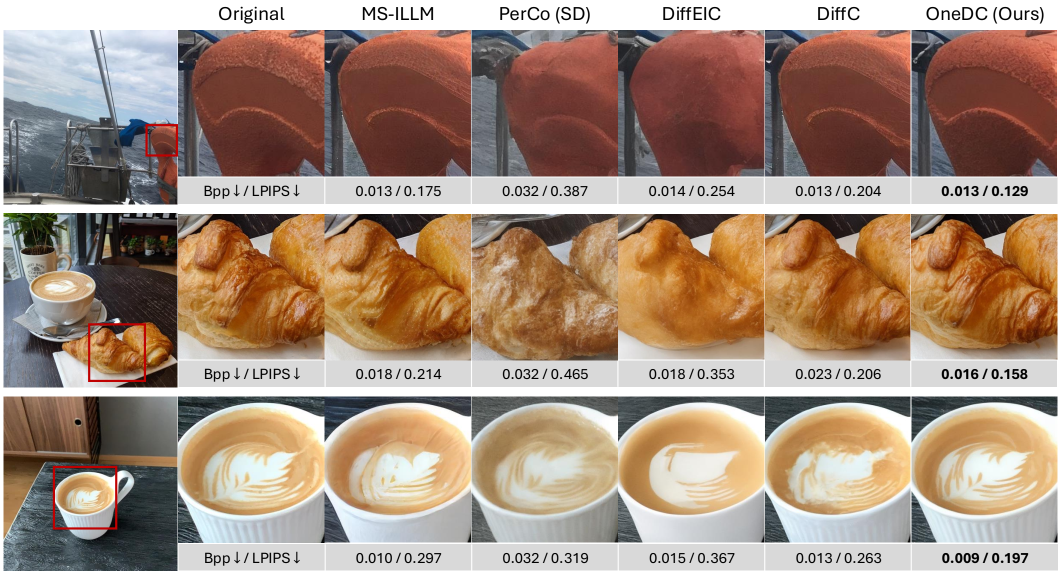


Figure 5: Visual examples on the CLIC2020 test set. Zoom in for better view.

with 4 quantization levels each, yielding an equivalent codebook size of 16,384. At a 64× spatial downsampling, this design results in a bitrate of just 0.0034 bpp, incurring very low transmission overhead. To substitute for text embeddings, we introduce a semantic decoder h_{sem} that transforms the quantized hyperprior $\hat{z}$ into a semantic context $c \in \mathbb{R}^{B \times N \times D}$, where $N = H' \times W'$ is the flattened spatial shape and D is the embedding dimension. This context is injected into the cross-attention layers of the one-step diffusion U-Net, replacing the textual embeddings used in the original design. In each cross-attention layer, the input latent feature f_{in} serves as the query, while the semantic context c provides both keys and values:

$$f_{out} = \text{Softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V, \quad \text{where } Q = W_Q f_{in},\ K = W_k c,\ V = W_v c \tag{1}$$

where d_k denotes the dimensionality of the key vectors and f_{out} is the output feature of the cross-attention layer. This design enables every spatial location in the latent to attend adaptively to semantically relevant tokens, strengthening content alignment. Compared with text conditioning, hyperprior-based guidance offers three advantages: (1) As shown in Fig. 2, conditioning the one-step diffusion model on categorical hyperprior tokens yields more faithful reconstructions than using text prompts, validating their effectiveness as semantic guidance. (2) The 64× downsampled hyperprior provides a large receptive field while retaining spatial locality, thereby offering a more balanced global–local semantics compared to purely global text embeddings. (3) Unlike separately trained text encoders [49], our approach enables end-to-end optimization of semantic guidance, allowing the hyperprior to adapt jointly with the diffusion model for improved content expression.

3.2.2 Semantic Distillation for Hyperprior

While the hyperprior effectively captures semantic cues for diffusion guidance, its representation capability can be further improved. In particular, we propose a semantic distillation mechanism aimed at fully tapping the potential of the hyperprior branch. This distillation guides the hyperprior encoder to better capture visual content and improve the decoding quality of the semantic decoder h_{sem}. Specifically, we transfer knowledge from a pretrained high-capacity generative tokenizer [15] to the hyperprior codec via an auxiliary distillation task. This choice is motivated by the structural similarity between the categorical hyperprior codec and the generative tokenizers that are known to produce semantically rich discrete representations [68]. By leveraging this compatibility, we use the tokenizer as a teacher to guide the hyperprior toward more effective semantic encoding.

Following GLC [22], we introduce a transformer-based predictor P_{aux} [72], trained to predict discrete token labels from the hyperprior-derived semantics c. The ground truth labels I_{gt} are obtained from a pretrained tokenizer encoder E_{aux}. Since the tokenizer is optimized for perceptual reconstruction, its objectives align well with OneDC, making it an effective teacher. Moreover, the smaller information

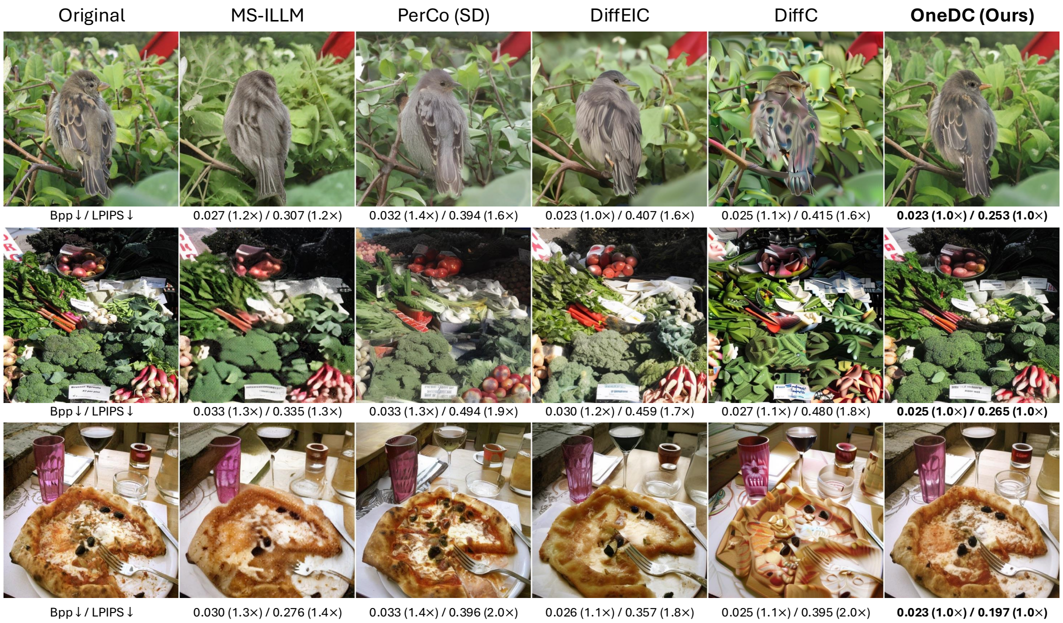


Figure 6: Visual examples on the MS-COCO 30K dataset. Zoom in for better view.

bottleneck in the hyperprior branch naturally filters out redundant information, distilling the most salient semantic features from the teacher. The distillation is supervised via cross-entropy (CE) loss:

$$I_{gt} = VQ(E_{aux}(x)), \quad L_{aux} = CE(I_{gt}, P_{aux}(c)) \tag{2}$$

Both P_{aux} and E_{aux} are used only during training, introducing no inference overhead. As shown in Fig.2, our semantic distillation enhances content fidelity, and ablation results in Section4.3 further validate its effectiveness in improving final reconstruction quality.

3.3 Training Strategy

To enhance training efficiency and stability, we adopt a two-stage training strategy, inspired by prior works [41, 44]. As shown in Fig. 4, stage I mainly trains the compression module while stage II focuses on fine-tuning the one-step diffusion model for better reconstruction quality. In both stages, the one-step diffusion model is adapted via LoRA layers [21], allowing fast convergence for the compression task while preserving rich generative priors.

Stage I: Pixel-domain Compression Learning This stage aims to: (1) train the codec to compress images into compact latent representations with high-fidelity reconstruction, (2) embed semantic information into the hyperprior via distillation, and (3) adapt the one-step diffusion model to synthesize fine-grained details on the decoder side. The total loss is defined as:

$$L_{stageI} = L_{recon} + \lambda R + \alpha L_{aux}, \quad \text{where } L_{recon} = L_1(x, \hat{x}) + L_{perceptual}(x, \hat{x}) \tag{3}$$

We use LPIPS [70] as the perceptual loss and L_1 as the pixel loss. R is the bitrate loss from the quad-tree-based spatial entropy model [30, 31], and L_{aux} is the semantic distillation loss (Eq. 2). The hyperparameter λ controls the compression ratio and α is the weight of semantic distillation loss.

Stage II: Hybrid-domain Perceptual Learning. This stage fine-tunes the one-step diffusion model to enhance perceptual realism without degrading fidelity. The latent compression module is fixed to improve training stability [41]. Prior work has shown that pixel-level optimization alone is insufficient for optimal perceptual quality [22]. Thus, we adopt a joint training strategy in both pixel and latent domains. Specifically, we integrate the diffusion-based distillation method from [66], which transfers generation knowledge from a pretrained multi-step diffusion teacher. The training also involves both perceptual supervision from the pixel domain and adversarial alignment in the latent domain, which ensures fidelity by aligning reconstruction with the original image. The total loss is:

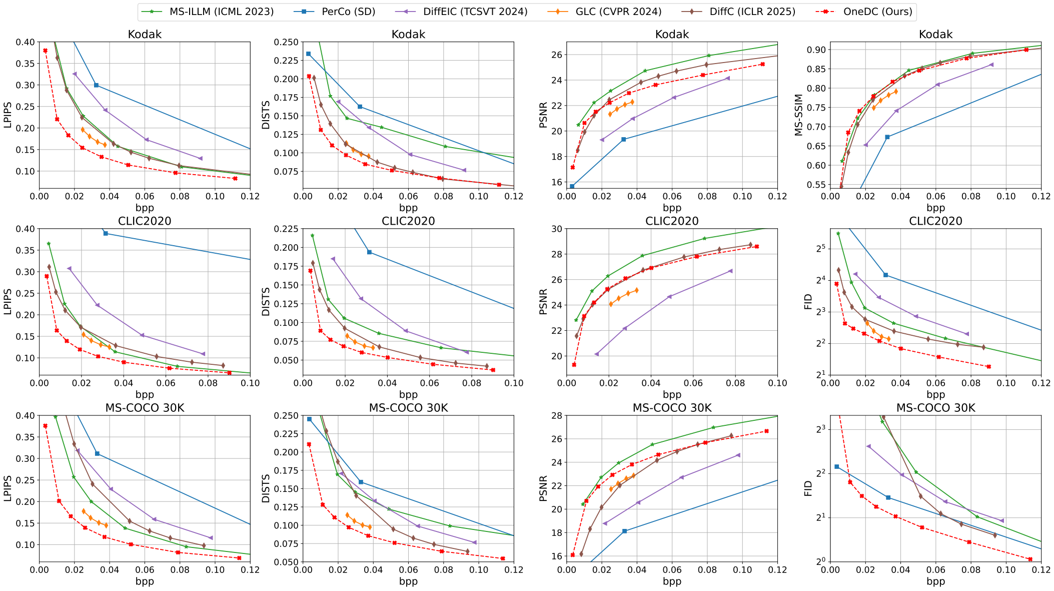


Figure 7: Rate–distortion curves under the *full-resolution* setting. Zoom in for better view.

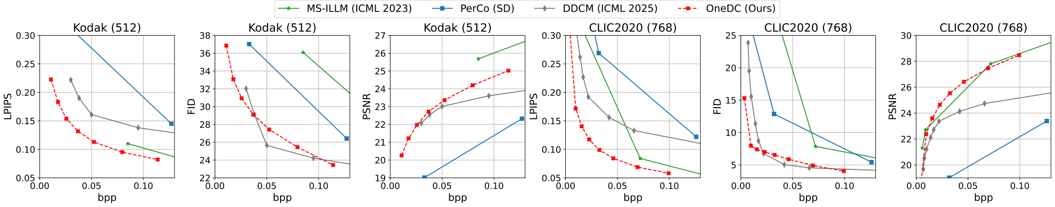


Figure 8: Rate–distortion curves under the *resize & center-crop* on Kodak 512×512 and CLIC2020 768×768. Zoom in for better view.

$$L_{stageII} = L_{distill} + \beta L_{recon} + \gamma L_{adv}, \quad \text{where:} \tag{4}$$

$$L_{distill} = \mathbb{E}_{t,\tilde{y}'_t}[\epsilon_{fake}(\tilde{y}'_t, t) - \epsilon_{real}(\tilde{y}'_t, t)], \quad L_{adv} = \mathbb{E}_{t,\tilde{y}'_t}[-Disc(\epsilon_{fake}(\tilde{y}'_t, t), t)] \tag{5}$$

Here, ϵ_{real} and ϵ_{fake} denote the real and fake score networks in the teacher model, and $\tilde{y}'_t$ is the noised latent at timestep t. $Disc$ refers to the discriminator. β and γ balance the pixel and adversarial losses. Additional implementation and loss details are provided in the supplementary material.

4 Experiment

4.1 Implementation and Evaluation Settings

Model and Training. Our generator adopts the U-Net architecture from the Stable Diffusion 1.5 (SD1.5) [50] and is initialized with weights from a pretrained one-step text-to-image task [66]. In Stage I, we use the pretrained tokenizer from MaskGIT [10] for semantic distillation. In Stage II, we employ the multi-step SD1.5 pretrained model as the teacher for diffusion distillation. To improve high-resolution adaptability, we randomly crop patches of size 512 or 1024 during training. Models are optimized using AdamW [38]. Additional settings are provided in the supplementary material.

Compared methods. We compare our OneDC with generative image codecs at a low-bitrate scenario. The compared methods include the vanilla VAE-based generative codec MS-ILLM [44] and GLC [22], and the recent diffusion-based methods: PerCo (SD) [28], DiffEIC [33], DiffC [57] (SD2.1-based) and DDCM [46]. For most methods, we evaluate at *full-resolution*, consistent with previous implementations in generative neural codecs [41, 44, 22]. For some diffusion-based methods like DDCM, we additionally provide results under the *resize & center-crop* setting, following DDCM protocol [46] for fair comparison. Further details are provided in the supplementary material.

Table 1: Ablation studies with BD-Rate (%) ↓.

Settings	CLIC2020	
	DISTS	FID
Semantic guidance		
No guidance	44.0	45.1
Text guidance	24.2	36.3
Hyperprior guidance	20.7	24.3
Hyperprior + Sem. Distil.→ **Ours**	0.00	0.00
Loss variation		
Pixel-domain only	11.4	51.8
Latent-domain only	60.7	37.1
Hybrid-domain → **Ours**	0.00	0.00

Table 2: Comparison of coding time and BD-Rate (%) ↓.

Methods	Times (s)		MS-COCO 30K		
	Enc.	Dec.	LPIPS	DISTS	FID
VAE-based					
MS-ILLM	0.14	0.17	138.3	253.0	478.4
Multi-step diffusion					
DiffEIC	0.32	12.4	305.0	239.1	341.0
PerCo (SD)	0.58	8.80	538.8	345.8	59.6
DiffC	3.9∼15.6	6.9∼10.8	234.0	196.1	690.9
One-step diffusion					
OneDC → **Ours**	0.15	0.34	0.00	0.00	0.00

Notes: Metrics in the tables denote those used for BD-Rate calculation [6]; **Ours** is the anchor (0.00%).

Evaluation datasets and metrics. We evaluate OneDC on several datasets, including Kodak [16], CLIC2020 test set [55], and MS-COCO 30K [35]. Reconstruction fidelity is evaluated using perceptual metrics LPIPS [24] and DISTS [13], along with the traditional metrics PSNR and MS-SSIM [60], while generative realism is measured by the no-reference perceptual metric FID [19]. Bitrate saving is measured by BD-Rate [6]. It is worth noting that FID is computed on overlapping 256×256 patches at *full-resolution* CLIC2020, following previous practice [44, 22]. For the MS-COCO 30K dataset with 512×512 images, we evaluate FID on full-images, consistent with [9, 28]. However, large patch leads to unreliable FID on the small test set (Kodak with 24 images) [9]. To address this, under the *resize & center-crop* setting we follow the DDCM [46] and compute FID on overlapping 64×64 patches for Kodak. More details are provided in the supplementary material.

4.2 Main Results

Quantitative Evaluation. Fig. 7 presents the *full-resolution* comparison between OneDC and existing methods across multiple distortion metrics. Across all datasets, including high-resolution CLIC2020, low-resolution Kodak and MS-COCO 30K, OneDC consistently achieves the best reference-based perceptual fidelity, as measured by LPIPS and DISTS. Compared with the previous SOTA, multi-step diffusion-based DiffC, OneDC achieves **55.27%** bitrate saving on Kodak and **54.60%** on CLIC2020 in terms of LPIPS, demonstrating superior compression efficiency. On traditional metrics such as PSNR, OneDC also exhibits competitive performance. For generation realism, OneDC attains lower FID than all multi-step diffusion baselines when FID is evaluated on the CLIC2020 test set. On the MS-COCO 30K dataset, OneDC provides a **39.55%** bitrate saving over PerCo (SD), previously the highest-realism method on this dataset. Such result further underscoring our method's effectiveness. Furthermore, Fig. 8 reports results under the *resize & center-crop* setting: OneDC achieves better LPIPS and PSNR than DDCM while maintaining competitive realism, demonstrating strong fidelity-oriented perceptual compression. In conclusion, OneDC delivers faithful and perceptually compelling reconstructions across a wide range of resolutions and content complexities, despite relying on a single sampling step. These evaluation results support our hypothesis that iterative sampling is unnecessary for high-quality diffusion-based generative compression.

Qualitative Evaluation. Fig. 5 presents visual comparisons on the CLIC2020 test set (*full-resolution*). At the lowest bitrate, OneDC produces the most natural and faithful reconstructions with sharp high-frequency details, while MS-ILLM yields noticeably blurred textures and multi-step diffusion-based methods (DiffEIC, PerCo, and DiffC) introduce structural distortions and inconsistent details. Similar trends are observed on the MS-COCO 30K dataset (Fig. 6), where MS-ILLM remains blurry and multi-step diffusion methods struggle to preserve visual fidelity. These qualitative results, together with the quantitative comparisons, highlight the SOTA performance of the proposed OneDC. Additional visual examples are provided in the supplementary material.

4.3 Ablation Studies and Efficiency Evaluation

We conduct ablation studies to investigate the effectiveness of semantic guidance and hybrid-domain perceptual learning. Models are evaluated on the CLIC2020 test set with BD-Rate [6].

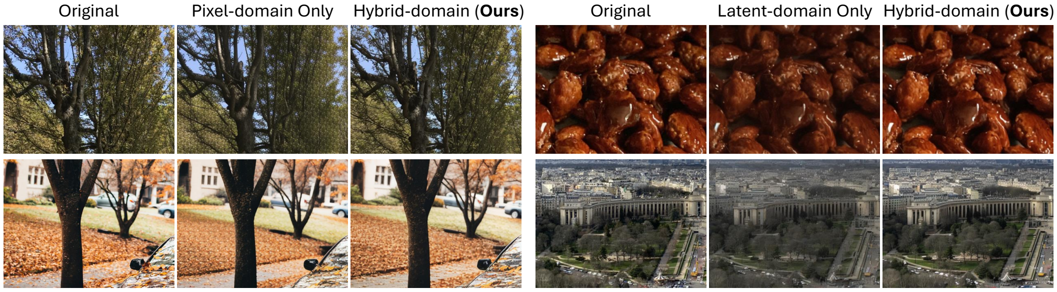


Figure 9: Visual examples of the ablation study in hybrid-domain perceptual learning. No latent-domain training leads artifacts (Left), while no pixel-domain supervision leads color shift (Right).

Semantic guidance. We first validate the importance of semantic guidance and then evaluate the effectiveness of our proposed semantic distillation for the hyperprior codec. As shown in Table 1, removing semantic guidance ("No Guidance") significantly degrades reconstruction quality—by up to 40%—highlighting its critical role in our one-step diffusion model. While text-based guidance improves performance, it underperforms the hyperprior guidance on high-resolution CLIC2020 images. This suggests that the hyperprior's locally captured semantics offer a more precise and spatially aligned representation of the image content. Further applying semantic distillation to the hyperprior ("Hyperprior + Sem. Distill.") leads to additional gains, indicating that transferring prior knowledge from a pretrained tokenizer enhances the semantic capability of our hyperprior branch.

Hybrid-domain Perceptual Learning. We evaluate the effectiveness of hybrid pixel-latent domain training in Stage II, as summarized in Table 1. (1) Using the pixel-domain loss only (*No* $L_{distill}$ and L_{adv}) leads to notable drops in both FID and DISTS scores. Visual examples in Fig. 9-Left show grid-like artifacts, reflecting a significant decline in perceptual quality. This underscores the importance of diffusion distillation in enforcing distributional alignment for realistic reconstruction. (2) Using the latent-domain loss only (*No* L_{recon}) also results in consistent degradation across all metrics, accompanied by visible color shifts in the reconstructions (Fig. 9-Right). This highlights the importance of pixel-level supervision in image compression, which emphasizes accurate reconstruction—unlike the pure image generation task that focuses solely on visual realism.

Operation Efficiency. We evaluate the coding times of different methods on 1024×1024 images using an A100 GPU. These runtime results are presented alongside BD-Rate results on the MS-COCO 30K dataset to offer a comprehensive view of operational efficiency. As shown in Table 2, OneDC achieves a significant decoding speed advantage—over $20\times$ faster—compared to multi-step diffusion-based methods, while still maintaining impressive reconstruction quality. Although OneDC is slightly slower than the pure VAE-based MS-ILLM due to its larger model, this added capacity is crucial for preserving perceptual quality under extreme compression, as evidenced by its superior DISTS and FID performance. Overall, these results underscore the effectiveness of OneDC in balancing fidelity, realism, and efficiency at ultra-low bitrates.

5 Conclusion

We present OneDC, a novel one-step diffusion-based generative image codec designed for high-quality and efficient compression under ultra-low bitrate scenarios. OneDC integrates a latent compression module for efficient coding and a one-step diffusion generator for fast reconstruction. To provide the critical semantic guidance for the one-step diffusion model and address the limitations of textual guidance, we leverage the hyperprior as a substitute. Furthermore, we improve the semantic capability of the hyperprior through a distillation mechanism. To enhance training efficiency and stability, we adopt a two-stage strategy: Stage I focuses on compression training through pixel-domain supervision, while Stage II refines reconstruction quality via hybrid-domain perceptual learning. Extensive experiments demonstrate that OneDC achieves SOTA perceptual quality with fast decoding, highlighting the strong potential of one-step diffusion models in generative image compression.

Limitation. Although OneDC offers substantial speedups over multi-step diffusion methods, its decoding speed does not meet real-time requirements. In future work, we plan to explore model distillation and architectural optimization to enhance efficiency.

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, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[3] 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.

[4] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *6th International Conference on Learning Representations, ICLR*, 2018.

[6] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *ITU SG16 Doc. VCEG-M33*, 2001.

[7] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pages 675–685. PMLR, 2019.

[8] 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. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

[9] 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.

[10] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948, 2020.

[12] Trung Dao, Thuan Hoang Nguyen, Thanh Le, Duc Vu, Khoi Nguyen, Cuong Pham, and Anh Tran. Swiftbrush v2: Make your one-step diffusion model better than its teacher. In *European Conference on Computer Vision*, pages 176–192. Springer, 2024.

[13] 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.

[14] Linwei Dong, Qingnan Fan, Yihong Guo, Zhonghao Wang, Qi Zhang, Jinwei Chen, Yawei Luo, and Changqing Zou. Tsd-sr: One-step diffusion with target score distillation for real-world image super-resolution. *arXiv preprint arXiv:2411.18263*, 2024.

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[16] Rich Franzen. Kodak PhotoCD dataset, 1999. URL `http://r0k.us/graphics/kodak/`.

[17] 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.

[18] Jinpei Guo, Zheng Chen, Wenbo Li, Yong Guo, and Yulun Zhang. Compression-aware one-step diffusion model for jpeg artifact removal. *arXiv preprint arXiv:2502.09873*, 2025.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[22] 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.

[23] Wei Jiang, Jiayu Yang, Yongqi Zhai, Peirong Ning, Feng Gao, and Ronggang Wang. Mlic: Multi-reference entropy model for learned image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7618–7627, 2023.

[24] 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.

[25] Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling diffusion models into conditional gans. In *European Conference on Computer Vision*, pages 428–447. Springer, 2024.

[26] Sohwi Kim and Tae-Kyun Kim. Tddsr: Single-step diffusion with two discriminators for super resolution. *arXiv preprint arXiv:2410.07663*, 2024.

[27] 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, 2024.

[28] Nikolai Körber, Eduard Kromer, Andreas Siebert, Sascha Hauke, Daniel Mueller-Gritschneder, 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`.

[29] 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.

[30] 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 (CVPR)*, pages 22616–22626, June 2023.

[31] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 17-21, 2024*, 2024.

[32] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[33] 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.

[34] Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Ajmal Mian. Diffusion-based extreme image compression with compressed feature initialization. *arXiv preprint arXiv:2410.02640*, 2024.

[35] 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.

[36] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14388–14397, 2023.

[37] 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.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[39] Lei Lu, Yanyue Xie, Wei Jiang, Wei Wang, Xue Lin, and Yanzhi Wang. Hybridflow: Infusing continuity into masked codebook for extreme low-bitrate image compression. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3010–3018, 2024.

[40] Qi Mao, Tinghan Yang, Yinuo Zhang, Zijian Wang, Meng Wang, Shiqi Wang, Libiao Jin, and Siwei Ma. Extreme image compression using fine-tuned vqgans. In *2024 Data Compression Conference (DCC)*, pages 203–212, 2024. doi: 10.1109/DCC58796.2024.00028.

[41] 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.

[42] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

[43] 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.

[44] 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.

[45] Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7807–7816, 2024.

[46] Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models. *arXiv preprint arXiv:2502.01189*, 2025.

[47] OpenAI. Gpt-4o. `https://openai.com/index/gpt-4o`, 2024. Accessed: 2025-05-05.

[48] Linfeng Qi, Zhaoyang Jia, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Generative latent coding for ultra-low bitrate image and video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[51] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[53] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023.

[54] Yanke Song, Jonathan Lorraine, Weili Nie, Karsten Kreis, and James Lucas. Multi-student diffusion distillation for better one-step generators. *arXiv preprint arXiv:2410.23274*, 2024.

[55] George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020), 2020. URL `http://www.compression.cc`.

[56] 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.

[57] Jeremy Vonderfecht and Feng Liu. Lossy compression with pretrained diffusion models. *arXiv preprint arXiv:2501.09815*, 2025.

[58] Guo-Hua Wang, Jiahao Li, Bin Li, and Yan Lu. EVC: Towards Real-Time Neural Image Compression with Mask Decay. In *International Conference on Learning Representations*, 2023.

[59] Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C Kot, and Bihan Wen. Sinsr: diffusion-based image super-resolution in a single step. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25796–25805, 2024.

[60] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003. doi: 10.1109/ACSSC.2003.1292216.

[61] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *Advances in Neural Information Processing Systems*, 37: 92529–92553, 2024.

[62] Tongda Xu, Jiahao Li, Bin Li, Yan Wang, Ya-Qin Zhang, and Yan Lu. PICD: Versatile Perceptual Image Compression with Diffusion Rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-25, 2024*, 2025.

[63] Naifu Xue, Qi Mao, Zijian Wang, Yuan Zhang, and Siwei Ma. Unifying generation and compression: Ultra-low bitrate image coding via multi-stage transformer. *arXiv preprint arXiv:2403.03736*, 2024.

[64] Naifu Xue, Zhaoyang Jia, Jiahao Li, Bin Li, Yuan Zhang, and Yan Lu. DLF: Extreme Image Compression with Dual-generative Latent Fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2025.

[65] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 22552–22562, 2023.

[66] 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.

[67] 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.

[68] Lijun Yu, Yong Cheng, Zhiruo Wang, Vivek Kumar, Wolfgang Macherey, Yanping Huang, David Ross, Irfan Essa, Yonatan Bisk, Ming-Hsuan Yang, et al. Spae: Semantic pyramid autoencoder for multimodal generation with frozen llms. *Advances in Neural Information Processing Systems*, 36:52692–52704, 2023.

[69] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[71] Jing Zhao, Bin Li, Jiahao Li, Ruiqin Xiong, and Yan Lu. A universal optimization framework for learning-based image codec. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(1):1–19, 2023.

[72] Shangchen Zhou, Kelvin Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. *Advances in Neural Information Processing Systems*, 35:30599–30611, 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We accurately provide the main contribution in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We present the limitation in the conclusion part.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not introduce new theoretical results; all theoretical foundations are attributed to their original sources through proper citation.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We thoroughly report all experimental settings, conditions, model architecture, and training details to ensure reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and models will be publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Details about the training/testing are included in our supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: There is no statistical significance tests in our paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computational resource usage is included in supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We strictly follow the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The potential societal impacts are presented in our supplementary material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The safeguard of codecs is ensured by the software and protocol level, not the codec itself.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We strictly follow the licenses and terms of the used code/data.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: There are no new assets included in our work.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing experiments included in our work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our core contribution and methodology are not influenced by LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Experiment

Evaluation of third-party models. We evaluate MS-ILLM [44] using the official checkpoints and fine-tune them with the provided code to support lower bitrates. For GLC [22], we report the results directly from its paper because the code and models are not publicly available at present. For DiffEIC [33] and DiffC [57], 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 [9], we use a publicly available reimplementation PerCo (SD) [28], as the original code has not been released. We also report the comparison with original PerCo by extract data from their paper (Fig. 10). For DDCM [46], 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 [41]. For the MS-COCO 30K dataset with 512×512 images, FID is evaluated on entire images, consistent with [9, 28]. 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 [46]. 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. 11. 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. 12. 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. 5 $\sim$ 10.

We also report traditional pixel-level distortion metrics (PSNR and MS-SSIM [60]) to provide a more comprehensive analysis, as shown in Fig. 13. At extremely low bitrates, optimizing for PSNR often suppresses high-frequency details, resulting in blurred reconstructions [41]. 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.16), CLIC2020—both full resolution (Figs.17) and 768×768 cropped (Fig.18), and MS-COCO 30K (Fig.19). 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 [66], replacing its textual semantic condition with the hyperprior features produced by our semantic hyperprior decoder h_{sem}. This enables reconstruction using only the hyperprior signal. Fig. 20 presents additional reconstruction results on the COCO2017 validation dataset [35], 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 4 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. 20), 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 3.

Table 3: 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 4: 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

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 [66] for the image reconstruction task. The training loss is defined as:

$$L_{stageI} = L_{recon} + \lambda R + \alpha L_{aux}, \quad \text{where} \;\; L_{recon} = L_1(x, \hat{x}) + L_{perceptual}(x, \hat{x}) \tag{6}$$

We use the L1 as the pixel-level loss and the LPIPS [24] 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 [17]. Training is performed on 4×A100 GPUs for 800,000 steps, using a three-stage learning rate schedule with AdamW [38]: 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 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:

$$L_{stageII} = L_{distill} + \beta L_{recon} + \gamma L_{adv}, \quad \text{where:} \tag{7}$$

$$L_{distill} = \mathbb{E}_{t,\tilde{y}'_t}[\epsilon_{fake}(\tilde{y}'_t, t) - \epsilon_{real}(\tilde{y}'_t, t)], \;\; L_{adv} = \mathbb{E}_{t,\tilde{y}'_t}[-Disc(\epsilon_{fake}(\tilde{y}'_t, t), t)] \tag{8}$$

Here, $L_{distill}$ represents the diffusion distillation loss [67], and L_{adv} is the adversarial loss in the latent space, following [66]. $Disc$ denotes the discriminator network, which takes the mid feature in the diffusion U-Net as the input [66]. 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

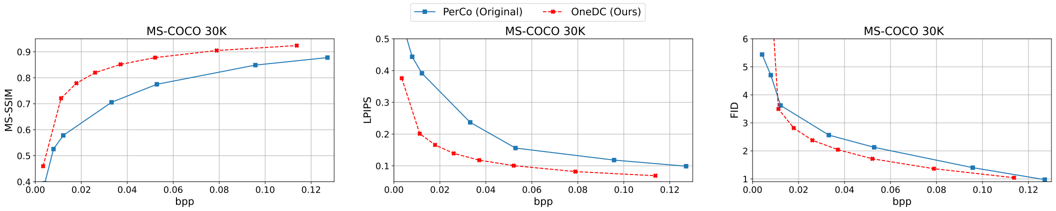


Figure 10: Comparison with original PerCo [9] on the MS-COCO 30K dataset.

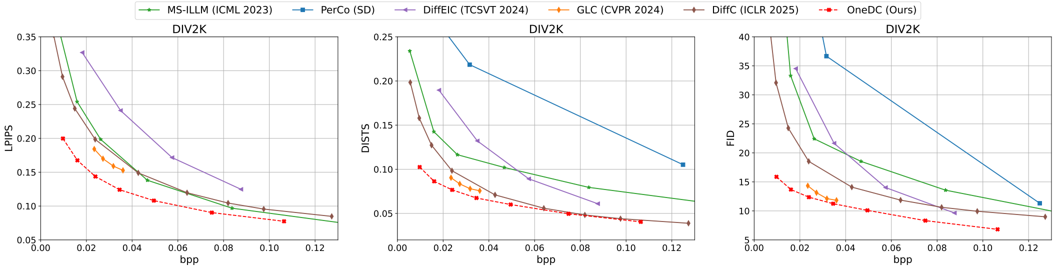


Figure 11: Rate–distortion curves on the DIV2K dataset at *full-resolution* setting.

set as follows: $\beta = 0.625$ balance the reconstruction and distillation terms, and $\gamma = 0.001$ for the adversarial loss (γ follows [66]).

Specifically, diffusion distillation [67] minimizes the expected Kullback-Leibler (KL) divergence between the time-dependent distributions of the target $p_{real,t}$ and the generator output $p_{real,t}$, 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:

$$\begin{aligned}
\nabla_\theta L_{distill} &= \mathbb{E}_{t,\tilde{y}_0}(\nabla_\theta \text{KL}(p_{fake,t}||p_{real,t})) && (9)\\
&= -\mathbb{E}_{t,\tilde{y}'_t}[(s_{real}(\tilde{y}'_t,t) - s_{fake}(\tilde{y}'_t,t))\frac{d\epsilon_\theta}{d\theta}] && (10)\\
&= \mathbb{E}_{t,\tilde{y}'_t}[(\epsilon_{fake}(\tilde{y}'_t,t) - \epsilon_{real}(\tilde{y}'_t,t))\frac{d\epsilon_\theta}{d\theta}] && (11)
\end{aligned}$$

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:

$$L_{fake} = \mathbb{E}_{t,\tilde{y}'_t}||\epsilon_{fake}(\tilde{y}'_t,t) - \tilde{y}_0||_2^2 \tag{12}$$

The improved version of diffusion distillation proposed in [66] 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:

$$L_{Disc} = \mathbb{E}_{t,x}[Disc(\epsilon_{fake}(\tilde{y}'_t,t)) - Disc(\epsilon_{fake}(E_{\text{VAE}}(x) + n_t, t))] \tag{13}$$

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:

$$L = L_{fake} + \sigma L_{Disc} \tag{14}$$

We follow [66] 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.

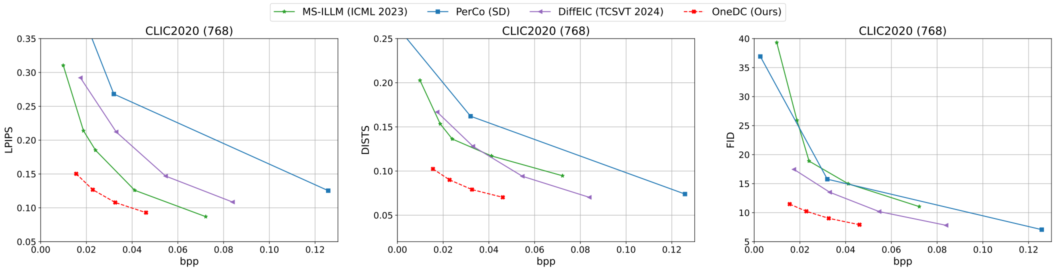


Figure 12: Rate–distortion curves on the CLIC2020 at *resize & center-crop, 256 FID* setting, with resized resolution 768.

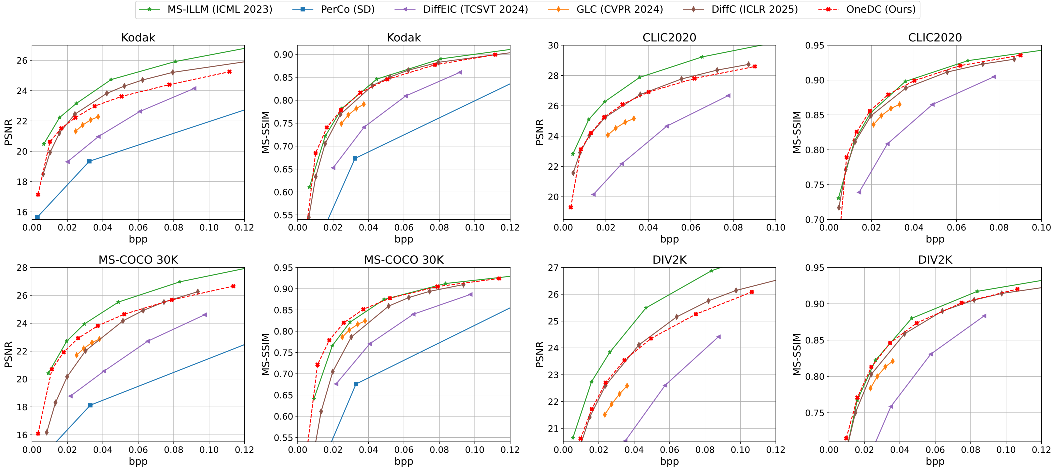


Figure 13: Comparison of methods measured by PSNR and MS-SSIM at *full-resolution* setting.

C Model Details

The overall architecture is shown in Fig.14. To ensure better alignment with the latent space of the diffusion model, we extract features from its pretrained VAE encoder. Additionally, inspired by DiffEIC[33], 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 [21] 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 ($g_a, h_{enc}, h_{ctx}, h_{sem}$ 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).

Further details of the semantic distillation mechanism are provided in Fig. 15. We adopt a Swin Transformer [37] 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.

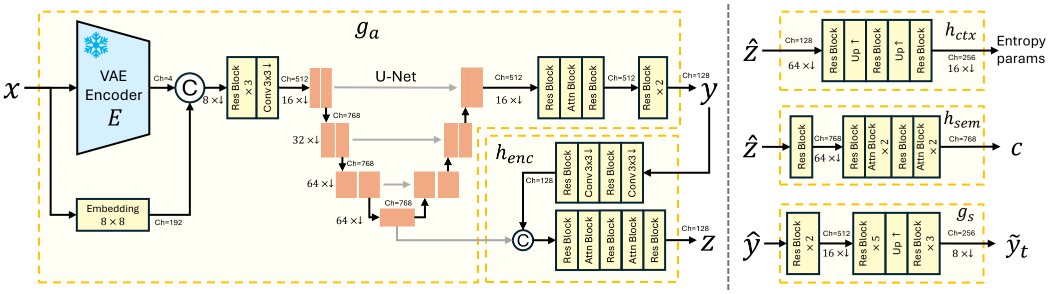


Figure 14: Details of our model architecture. The pratrained VAE encoder is from SD 1.5. For the U-Net used in g_a, we use the implementation from the diffusers library [56].

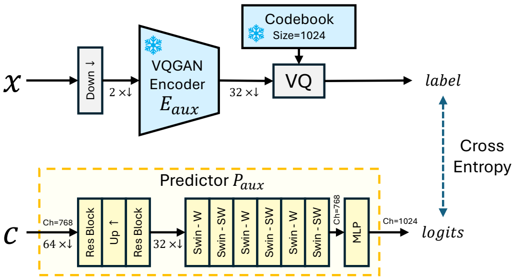


Figure 15: Details of the auxiliary code predictor module. For Swin Transformer block [37], we set window size to 16, head dim to 64. W means normal window, SW means shifted window.

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 [3] offers a potential path forward, enabling more transparent control over the trade-off between realism and fidelity.

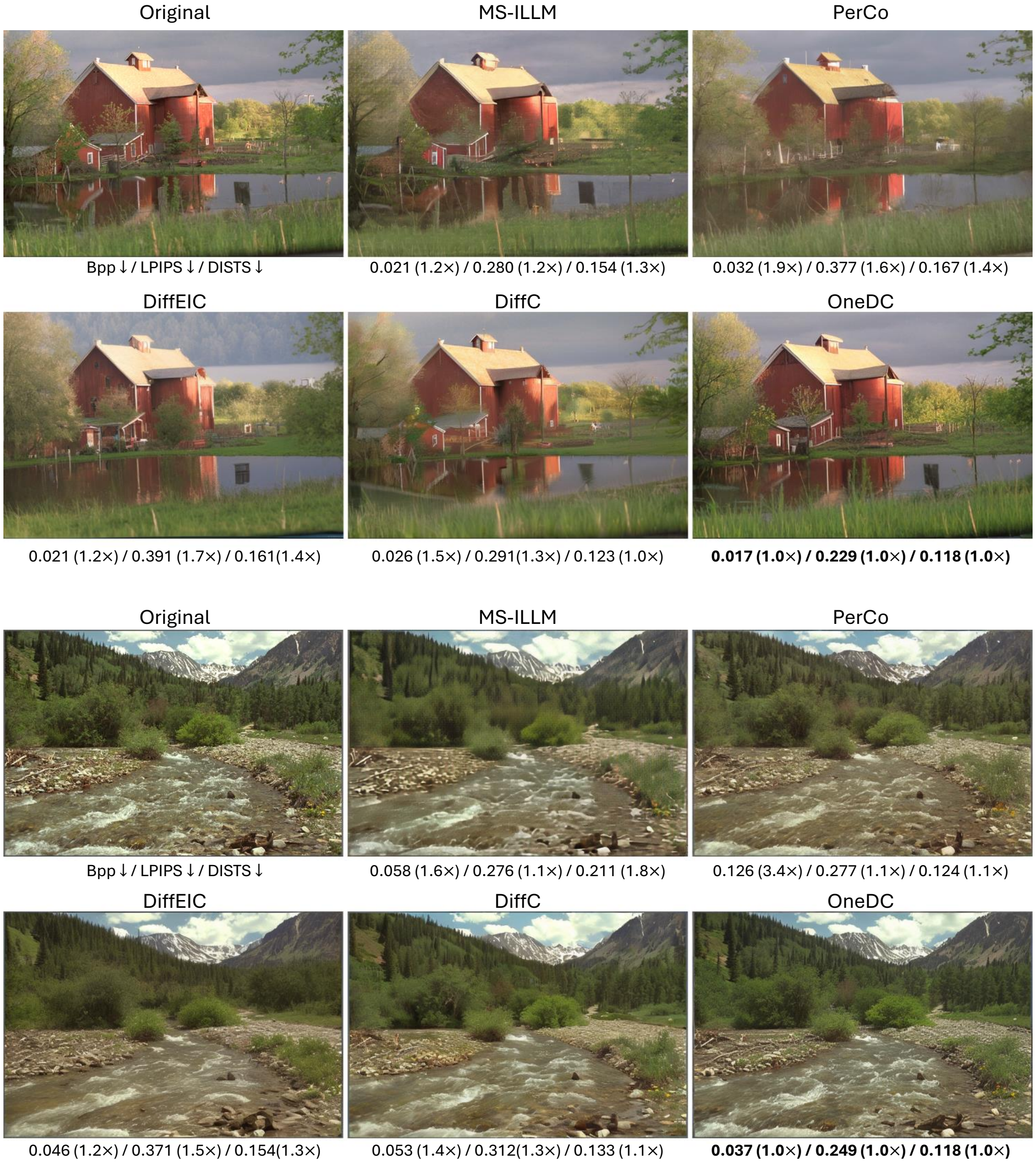


Figure 16: Visual results on the Kodak dataset. The VAE-based MS-ILLM exhibits noticeable artifacts. Compared with our OneDC, the previous SOTA multi-step diffusion codec DiffC requires at least $1.4\times$ bitrate on these two examples. Zoom in for better view.

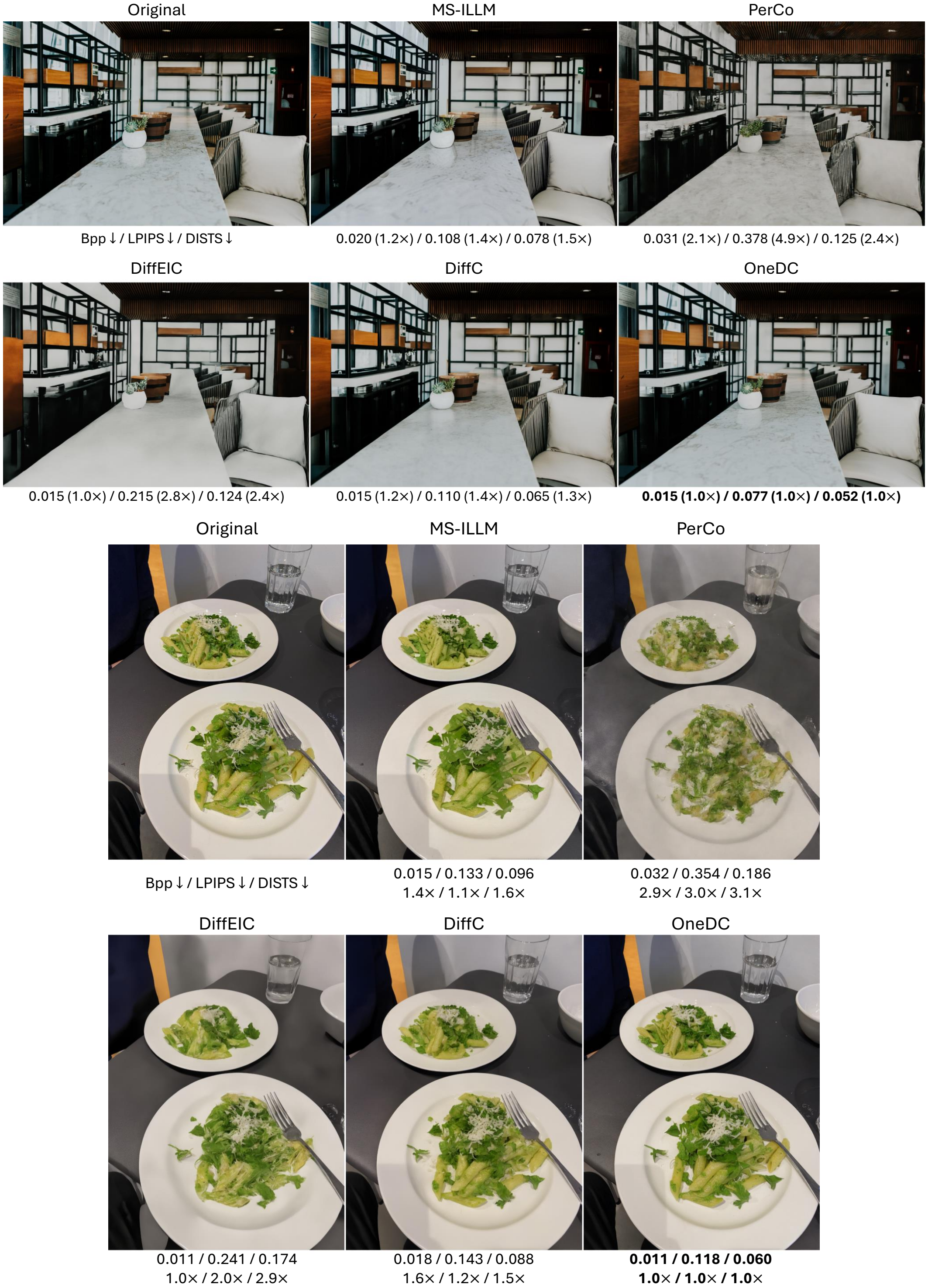


Figure 17: Visual results on the CLIC2020 test set. Zoom in for better view.

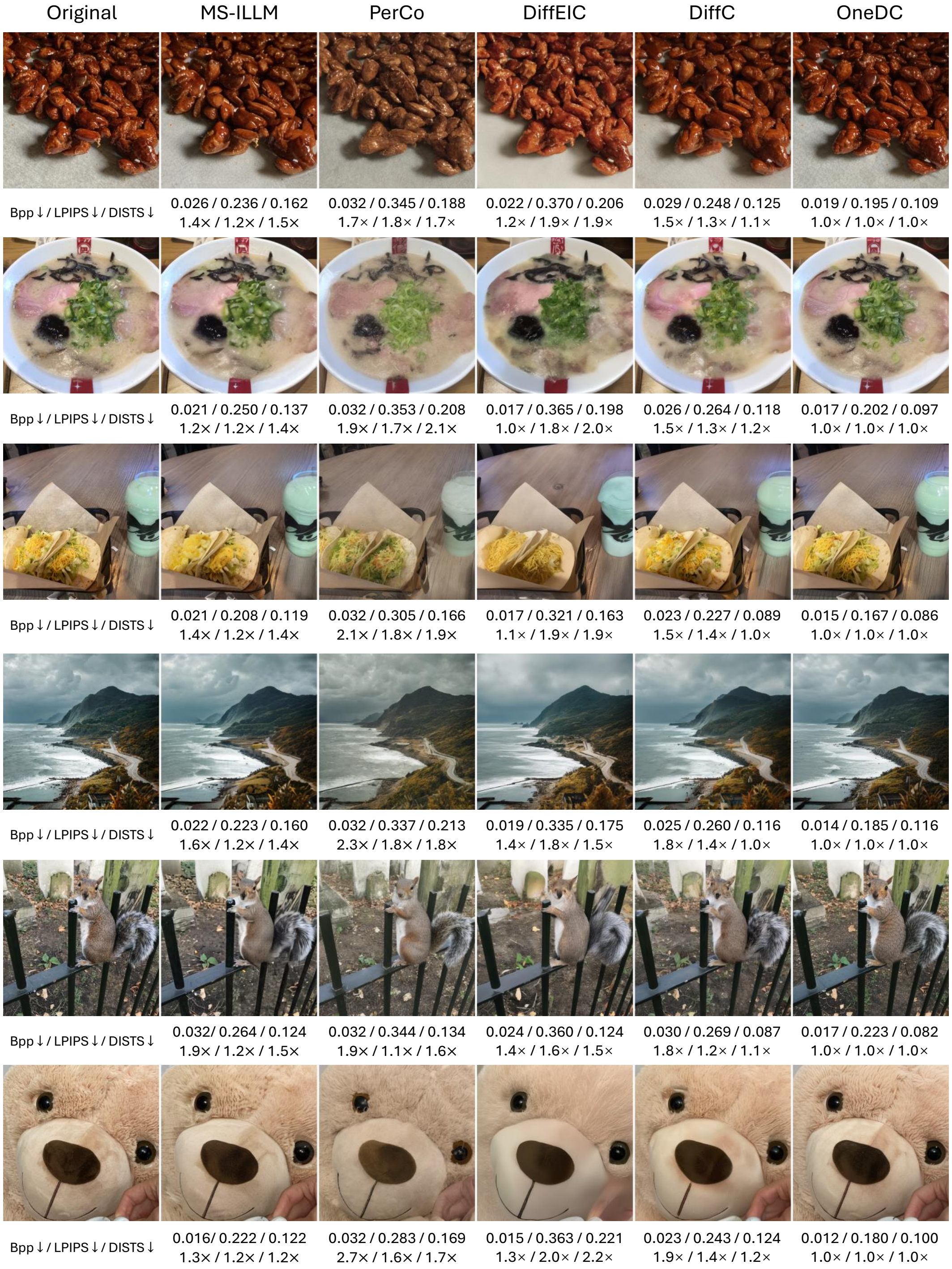


Figure 18: Visual results on the CLIC2020 test set (768×768). Compared with OneDC, the previous SOTA multi-step diffusion method DiffC requires ≥ 1.5 higher bitrate while still producing slightly lower reconstruction quality on these examples. Zoom in for better view.

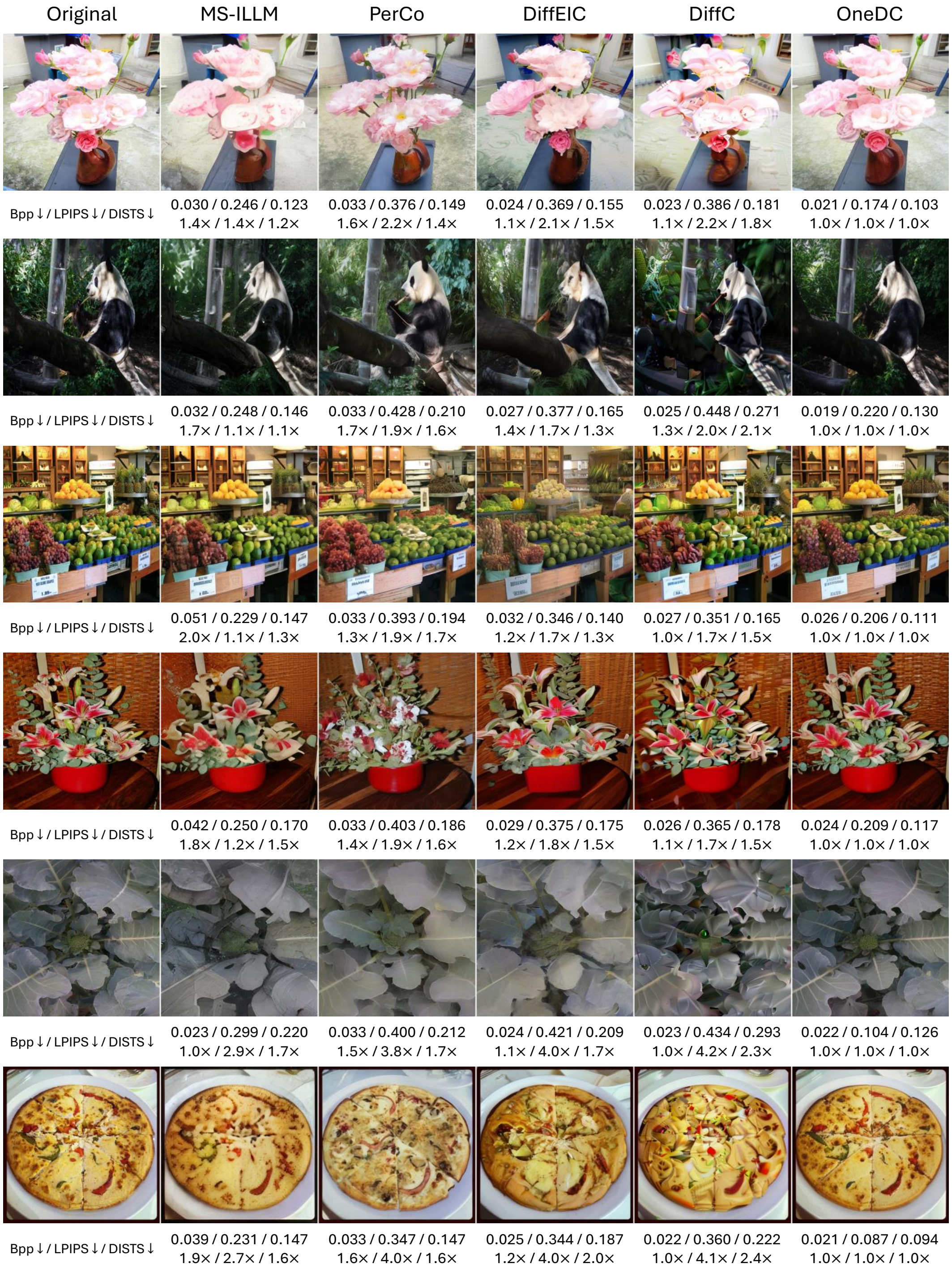


Figure 19: Visual results on the MS-COCO 30K dataset. PerCo, the previous best-performing method in terms of FID, requires ≥ 1.4 bitrate over OneDC, but still results in suboptimal fidelity. Zoom in for better view.

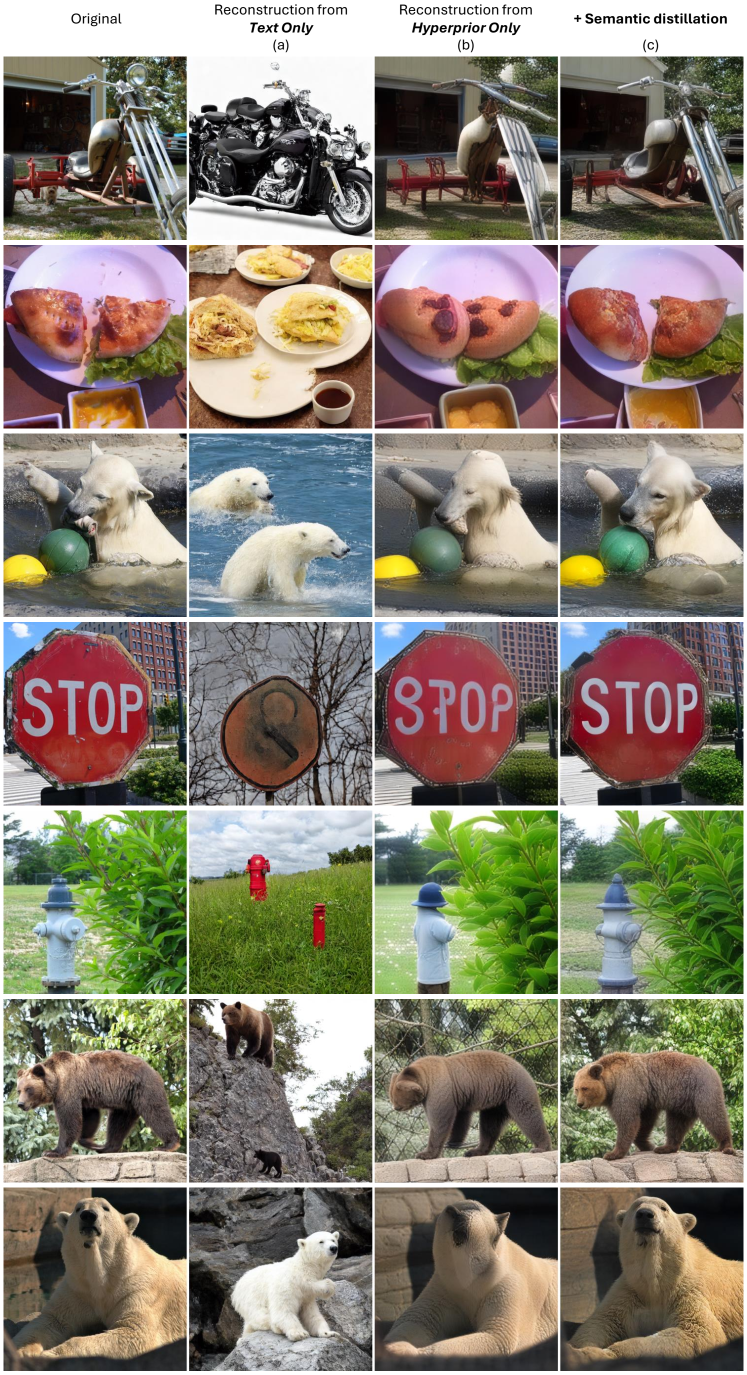


Figure 20: Reconstructions from different semantic guidance. (a) Text prompts struggle to capture complex visual semantics, resulting in severe distortions when using a pretrained text-to-image one-step diffusion model [66]. (b) We finetune the model [66] for hyperprior-to-image generation. Hyperprior guidance yields more faithful reconstructions. (c) Our proposed semantic distillation further improves object-level accuracy.

Table 5: 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 6: 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 7: 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 8: 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 9: 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 10: 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