Diff-PCC: Diffusion-based Neural Compression for 3D Point Clouds

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Stable diffusion networks have emerged as a groundbreaking development for their ability to produce realistic and detailed visual content. This characteristic renders them ideal decoders, capable of producing high-quality and aesthetically pleasing reconstructions. In this paper, we introduce the first diffusion-based point cloud compression method, dubbed Diff-PCC, to leverage the expressive power of the diffusion model for generative and aesthetically superior decoding. Different from the conventional autoencoder fashion, a dual-space latent representation is devised in this paper, in which a compressor composed of two independent encoding backbones is considered to extract expressive shape latents from distinct latent spaces. At the decoding side, a diffusion-based generator is devised to produce high-quality reconstructions by considering the shape latents as guidance to stochastically denoise the noisy point clouds. Experiments demonstrate that the proposed Diff-PCC achieves state-of-the-art compression performance (e.g., 7.711 dB BD-PSNR gains against the latest G-PCC standard at ultra-low bitrate) while attaining superior subjective quality. Source code will be made publicly available.

1 Introduction

Point clouds, composed of numerous discrete points with coordinates (x, y, z) and optional attributes, offer a flexible representation of diverse 3D shapes and are extensively applied in various fields such as autonomous driving [8], game rendering [35], robotics [7], and others. With the rapid advancement of point cloud acquisition technologies and 3D applications, effective point cloud compression techniques have become indispensable to reduce transmission and storage costs.

1.1 Background

Prior to the widespread adoption of deep learning techniques, the most prominent traditional point cloud compression methods were the G-PCC [39] and V-PCC [40] proposed by the Moving Picture Experts Group(MPEG). G-PCC compresses point clouds by converting them into a compact tree structure, whereas V-PCC projects point clouds onto a 2D plane for compression. In recent years, numerous deep learning-based methods have been proposed [50, 45, 11, 12, 7, 30, 46, 14, 42], which primarily employ the Variational Autoencoder (VAE) [1, 2] architecture. By learning a prior distribution of the data, the VAE projects the original input into a higher-dimensional latent space, and reconstructs the latent representation effectively using a posterior distribution. However, previous VAE-based point cloud compression architectures still face recognized limitations: 1) Assuming a single Gaussian distribution $N(\mu, \sigma^2)$ in the latent space may prove inadequate to capture the intricate diversity of point cloud shapes, yielding blurry and detail-deficient reconstructions [56, 10]; 2) The Multilayer Perceptron (MLP) based decoders [50, 45, 11, 12, 46] suffer from feature homogenization, which leads to point clustering and detail degradations in the decoded point cloud surfaces, lacking the

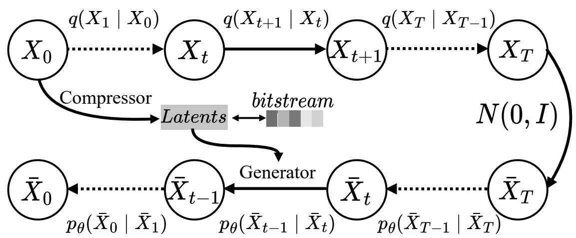

Figure 1: Diff-PCC pipeline. X_t and $\bar{X}_t$ represents the tth original point cloud and noisy point cloud, respectively; p refers to the forward process and q refers to the reverse process; $N(0, \boldsymbol{I})$ means the pure noise. Entropy model and arithmetic coding is omitted for a concise explanation.

ability to produce high-quality reconstructions. Recently, Diffusion models (DMs) [5] have attracted considerable attention in the field of generative modeling [34, 48, 41, 19] due to their outstanding performance in generating high-quality samples and adapting to intricate data distributions, thus presenting a novel and exciting opportunity within the domain of neural compression [33, 44, 25]. By generating a more refined and realistic 3D point cloud shape, DMs offer a distinctive approach to reduce the heavy dependence of reconstruction quality on the information loss of bottleneck layers.

1.2 Our Approach

Building on the preceding discussion, we introduce Diff-PCC, a novel lossy point cloud compression framework that leverages diffusion models to achieve superior rate-distortion performance with exceptional reconstruction quality. Specifically, to enhance the representation ability of simplistic Gaussian priors in VAEs, this paper devises a dual-space latent representation that employs two independent encoding backbones to extract complementary shape latents from distinct latent spaces. At the decoding side, a diffusion-based generator is devised to produce high-quality reconstructions by considering the shape latents as guidance to stochastically denoise the noisy point clouds. Experiments demonstrate that the proposed Diff-PCC achieves state-of-the-art compression performance (e.g., 7.711 dB BD-PSNR gains against the latest G-PCC standard at ultra-low bitrate) while attaining superior subjective quality.

1.3 Contribution

Main contributions of this paper are summarized as follows:

- We propose Diff-PCC, a novel diffusion-based lossy point cloud compression framework. To the best of our knowledge, this study presents *the first* exploration of diffusion-based neural compression for 3D point clouds.
- We introduce a dual-space latent representation to enhance the representation ability of the conventional Gaussian priors in VAEs, enabling the Diff-PCC to extract expressive shape latents and facilitate the following diffusion-based decoding process.
- We devise an effective diffusion-based generator to produce high-quality noises by considering the shape latents as guidance to stochastically denoise the noisy point clouds.

2 Related Work

2.1 Point Cloud Compression

Classic point cloud compression standards, such as G-PCC, employ octree[29] to compress point cloud geometric information. In recent years, inspired by deep learning methods in point cloud analysis[26, 27] and image compression[1, 2, 22], researchers have turned their attention to learning-based point cloud compression. Currently, point cloud compression methods can be primarily divided into two branches: voxel-based and point-based approaches. Voxel-based methods further branch into sparse convolution[36, 37, 38, 49, 51, 52] and octree[9, 24, 31]. Among them, sparse convolution derives from 2D-pixel representations but optimizes for voxel sparsity. On the other hand, octree-based methods, utilize tree structures to eliminate redundant voxels, representing only the occupied ones. Point-based methods[11, 50, 45, 46] are draw inspiration from PointNet [26], utilizing symmetric operators (max pooling, average pooling, attention pooling) to handle permutation-invariant point clouds and capture geometric shapes. For compression, different quantization operations categorize point cloud compression into lossy and lossless types. In this paper, we focus on lossy compression to achieve higher compression ratios by sacrificing some precision in the original data.

2.2 Diffusion Models for Point Cloud

Recently, diffusion models have ignited the image generation field[58, 17, 32], inspiring researchers to explore their potential in point cloud applications. DPM[20] pioneered the introduction of diffusion models in this domain. Starting from DPM, PVD[57] combines the strengths of point cloud and voxel representations, establishing a baseline based on PVCNN. LION[47] employs two diffusion models to separately learn shape representations in latent space and point representations in 3D space. Dit-3D[23] innovates by integrating transformers into DDPM, directly operating on voxelized point clouds during the denoising process. PDR[21] employs diffusion model twice during the process of generating coarse point clouds and refined point clouds. Point·E[] utilizes three diffusion models for the following processes: text-to-image generation, image-to-point cloud generation, and point cloud upsampling. PointInfinity[13] utilizes cross-attention mechanism to decouple fixed-size shape latent and variable-size position latent, enabling the model to train on low-resolution point clouds while generating high-resolution point clouds during inference. DiffComplete[4] enhances control over the denoising process by incorporating ControlNet[53], achieving new state-of-the-art performances. These advancements demonstrate the promise of DMs in point cloud generation tasks, which motivates our exploring its applicability in point cloud compression. Our research objective is to explore the effective utilization of diffusion models for point cloud compression while preserving its critical structural features.

3 Method

Figure 1 illustrates the pipeline of the proposed Diff-PCC, which can also represent the general workflow of diffusion-based neural compression. A concise review for Denoising Diffusion Probabilistic Models (DDPMs) and Neural Network (NN) based point cloud compression is first provided in Sec. 3.1; The proposed Diff-PCC is detailed in Sec. 3.2.

3.1 Preliminaries

Denoising Diffusion Probabilistic Models (DDPMs) comprise two Markov chains of length T: diffusion process and denoising process. Diffusion process adds noise to clean data $\boldsymbol{x_0}$, resulting in a series of noisy samples $\{\boldsymbol{x_1}, \boldsymbol{x_2}...\boldsymbol{x_T}\}$. When T is large enough, $x_T \sim N(0, \boldsymbol{I})$. The denoising process is the reverse process, gradually removing the noise added during the diffusion process. We formulate them as follows:

$$q(\boldsymbol{x_1}, \cdots, \boldsymbol{x_T}|\boldsymbol{x_0}) = \prod_{t=1}^{T} q(\boldsymbol{x_t}|\boldsymbol{x_{t-1}}), \text{ where } q(\boldsymbol{x_t}|\boldsymbol{x_{t-1}}) = \mathcal{N}(\boldsymbol{x_t}; \sqrt{1-\beta_t}\boldsymbol{x_{t-1}}, \beta_t \boldsymbol{I}) \tag{1}$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x_0}, \cdots, \boldsymbol{x_{T-1}}|\boldsymbol{x_T}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x_{t-1}}|\boldsymbol{x_t}), \text{ where } p_{\boldsymbol{\theta}}(\boldsymbol{x_{t-1}}|\boldsymbol{x_t}) = \mathcal{N}(\boldsymbol{x_{t-1}}; \boldsymbol{\mu_\theta}(\boldsymbol{x_t}, t), \sigma_t^2 \boldsymbol{I}) \tag{2}$$

where β is a hyperparameter representing noise level. $t \sim \text{Unif}\{1, \ldots, T\}$ represents time step. Via reparameterization trick, we can sample from $q(\boldsymbol{x_t}|\boldsymbol{x_{t-1}})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x_{t-1}}|\boldsymbol{x_t})$ as following:

$$x_t = \sqrt{1-\boldsymbol{\beta_t}}x_{t-1} + \sqrt{\boldsymbol{\beta_t}}\boldsymbol{\epsilon} \tag{3}$$

$$x_{t-1} = \boldsymbol{\mu_\theta}(\boldsymbol{x}_t, t) + \sigma_t \boldsymbol{\epsilon} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x_t} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon_\theta}(\boldsymbol{x_t}, t)\right) + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}\epsilon \tag{4}$$

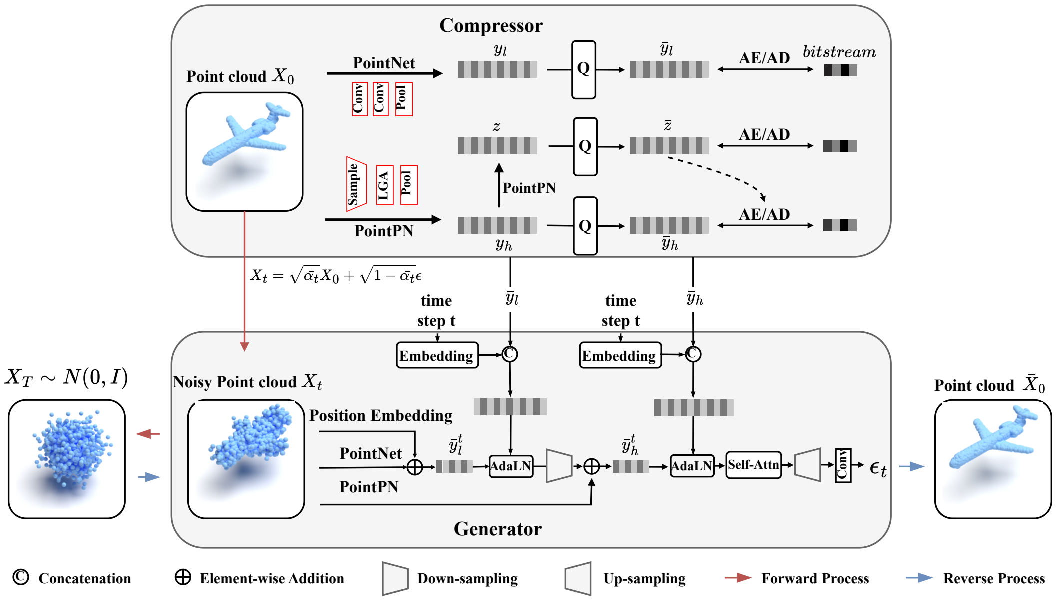

Figure 2: Detailed Structure of the Utilized Compressor and Generator. y_l and y_h refer to the low-frequency shape latent and high-frequency detail latent, respectively; z means hyperprior latent; Q refers to the quantization; AE and AD represents the arithmetic encoding and decoding.

where $\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, ϵ denotes random noise sampled from $\boldsymbol{N}(0, \boldsymbol{I})$. Note that $\boldsymbol{\epsilon_\theta}(\boldsymbol{x_t}, t)$ is a neural network used to predict noise during the denoising process, and $\boldsymbol{x_t}$ can be directly sampled via $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$.

DDPMs train the reverse process by optimizing the model parameters θ through noise distortion. The loss function $L(\theta, \boldsymbol{x}_0)$ is defined as the expected squared difference between the predicted noise and the actual noise, with the mathematical expression as follows:

$$L(\theta, \boldsymbol{x}_0) = \boldsymbol{E}_{t,\epsilon} ||\epsilon - \epsilon_\theta(\boldsymbol{x}_t, t)||^2 \tag{5}$$

3.2 DIFF-PCC

3.2.1 Overview

As shown in Fig. 2, two key components, i.e., compressor and generator, are respectively utilized in the diffusion process and denoising process. In Diff-PCC, the diffusion process is identified as the encoding, in which a compressor extracts latents from the point cloud and compresses latents into bitstreams; at the decoding side, the generator accepts the latents as a condition and gradually restoring point cloud shape from noisy samples.

3.2.2 Dual-Space Latent Encoding

Several research have demonstrated that a simplistic Gaussian distribution in the latent space may prove inadequate to capture the complex visual signals [56, 3, 6, 10]. Although previous works have proposed to solve these problems using different technologies such as non-gaussian prior [15] or coupling between the prior and the data distribution [10], these techniques may not be able to directly employed on neural compression tasks.

In this paper, a simple yet effective compressor is introduced, which composed of two independent encoding backbones to extract expressive shape latents from distinct latent spaces. Motivated by PointPN [55], which excels in capturing high-frequency 3D point cloud structures characterized by sharp variations, we design a dual-space latent encoding approach that utilizes PointNet to extract low-frequency shape latent and leverages PointPN to characterize complementary latent from high frequency domain. Let x be the original input point cloud, we formulate the above process as:

$$\{y_l, y_h\} = \{E_l(x), E_h(x)\} \tag{6}$$

where $y_l \in \mathbb{R}^{1\times C}$ and $y_h \in \mathbb{R}^{S\times C}$ represent the low-frequency and high-frequency latent features, respectively; E_l and E_h refer to the PointNet and PointPN backbones, respectively. Next, the quantization process Q is applied on the obtained features $\bar{y}_l$ and $\bar{y}_h$, i.e.,

$$\{\bar{y}_l, \bar{y}_h\} = \{Q(y_l), Q(y_h)\} \tag{7}$$

where function Q refers to the operation of adding uniform noise during training [1] and the rounding operation during test.

Then, fully factorized density model [1] and the hyperprior density model [2] are employed to fit the distribution of quantized features $\bar{y_l}$ and $\bar{y_h}$, respectively. Particularly, the hyperprior density model $p_\varphi(\bar{y}_h)$ can be described as:

$$p_\varphi(\bar{y}_h) = \left(N(\mu, \sigma^2) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right)\right)(\bar{y}_h) \tag{8}$$

where $\mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right)$ refers to the uniform noise ranging from $-\frac{1}{2}$ to $\frac{1}{2}$; $N(\mu, \sigma^2)$ refers to the normal distribution with expectation μ and standard deviation σ, which can be further estimated by a hyperprior encoder E_{hyper} and decoder D_{hyper}:

$$(\mu, \sigma^2) = D_{hyper}(\bar{z}) = D_{hyper}(Q(z)) = D_{hyper}(Q(E_{hyper}(y_h))) \tag{9}$$

In this way, a triplet containing quantized low-frequency feature $\bar{y}_l$, quantized high-frequency feature $\bar{y}_h$, and quantized hyperprior $\bar{z}$ will be compressed into three separate streams. Let $p(\cdot)$ and $p_{(\dots)}(\cdot)$ respectively represents the actual distribution and estimated distribution of latent features, then the bitrate $\mathcal{R}$ can be estimated as follows:

$$\mathcal{R} = \mathbb{E}_{\bar{y}_l\sim p(\bar{y}_l)}\left[-\log_2 p_\theta(\bar{y}_l)\right] + \mathbb{E}_{\bar{y}_h\sim p(\bar{y}_h)}\left[-\log_2 p_\varphi(\bar{y}_h)\right] + \mathbb{E}_{\bar{z}\sim p(\bar{z})}\left[-\log_2 p_\phi(\bar{z})\right] \tag{10}$$

3.2.3 Diffusion-based Generator

The generator takes noisy point cloud x_t at time t and necessary conditional information C as input. We hope generator to learn positional distribution F of x_t and fully integrate F with C to predict noise ϵ_t at time t. In this paper, we consider all information that could potentially guide the generator as conditional information, including time t, class label l, noise coefficient β_t, and decoded latent features ($\bar{y}_l$ and $\bar{y}_h$).

DiffComplete [4] uses ControlNet [54] to achieve refined noise generation. However, the denoiser of DiffComplete is a 3D-Unet, adapted from its 2D version [16]. This structure is not suitable for our method, because we directly deal with points, instead of voxels. We embraced this idea and specially designed a hierarchical feature fusion mechanism to adapt to our method. Note that 3D-Unet can directly downsample features F through 3D convolution with a stride greater than one. It is very complex for point-based methods to achieve equivalent processing. Therefore, we did not replicate the same structure as DiffComplete does, but directly used AdaLN to inject conditional information, formulated as:

$$AdaLN(F_{in}, C) = Norm(F_{in}) \odot Linear(C) + Linear(C) \tag{11}$$

where F_{in} denotes the original features in the Generator and C denotes the condition information.

Now we detail the structure: First, we need to exact the shape latent of noise point cloud x_t and we choose PointNet for structural consistency. However, in the early stages of the denoising process, x_t lacks a regular surface shape for the generator to learn. Therefore, we adopt the suggestion from PDR [23], adding positional encoding to each noise point so that the generator can understand the absolute position of each point in 3D space. Then we inject shape latent $\bar{y}_l$ from the compressor via ADaLN. We formulate the above process as:

$$F_{x_t} = PointNet(x_t) + PE(x_t) \tag{12}$$

$$F_{xt}^{'} = AdaLN(F_{x_t}, C) \tag{13}$$

Next, we need to fuse high-frequency features. We extract the local high-frequency features of x_t using PointPN and add them to F from the previous step, Then we inject the high-frequency features from the compressor via AdaLN. We use K-Nearest Neighbor (KNN) operation to partition locally and set the number of neighbor points to 8, which allows the generator to learn local details. We formulate the above process as:

$$F^{'} = PointPN(x_t) + FPS(F_{in}) \tag{14}$$

$$F_{out} = AdaLN(F^{'}, C) \tag{15}$$

After that, we use the self-attention mechanism to interact with information from different local areas. And through a feature up-sampling module, we generate features for n points. Finally, we output noise through a linear layer. We formulate the above process as:

$$F^{'} = SA(F_{in}) \tag{16}$$

$$F^{''} = UP(F^{'}) \tag{17}$$

$$\epsilon_t = Linear(F^{''}) \tag{18}$$

3.2.4 Training Objective

We follow the conventional rate-distortion trade-off as our loss function as follows:

$$\mathcal{L} = \mathcal{D} + \lambda\mathcal{R} \tag{19}$$

where $\mathcal{D}$ refers to the evaluated distortion; $\mathcal{R}$ represents bitrate as shown in Eq. 10; λ serves as the balance the distortion and bitrate. Specifically, a combined form of distortion $\mathcal{D}$ is used in this paper, which considers both intermediate noises (ϵ, $\bar{\epsilon}$) and global shapes (x_0, $\bar{x}_0$):

$$\mathcal{D} = \mathcal{D}_{MSE}(\epsilon, \bar{\epsilon}) + \gamma\mathcal{D}_{CD}(x_0, \bar{x}_0) \tag{20}$$

where $\mathcal{D}_{MSE}$ denotes the Mean Squared Error (MSE) distance; $\mathcal{D}_{CD}$ refers to the Chamfer Distance; γ means the weighting factor. Here, the overall point cloud shape is additively supervised under the Chamfer Distance $\mathcal{D}_{CD}(x_0, \bar{x}_0)$ to provide a global optimization. The following function is utilized to predict the reconstructed point cloud $\bar{x}_0$ in practice:

$$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta \left(x_t, t, c \right) \right) \tag{21}$$

where $\bar{\alpha}_t$ means the noise level; x_t refers to the noisy point cloud at time step t; ϵ_θ denotes the predicted noise from the generator; c represent the conditional information we inject into the generator.

4 Experiments

4.1 Experimental Setup

Datasets Based on previous work, we used ShapeNet as our training set, sourced from [20]. This dataset contains 51,127 point clouds, across 55 categories, which we allocated in an 8:1:1 ratio for training, validation, and testing. Each point cloud has 15K points, and following the suggestions from [28], we randomly select 2K points from each for training. Additionally, we also used ModelNet10 and ModelNet40 as our test sets, sourced from [43]. These datasets contain 10 categories and 40 categories respectively, totaling 10,582 point clouds. During training and testing, we perform individual normalization on the shape of each point cloud.

Baselines & Metric We compare our method with the state-of-the-art non-learning-based method: G-PCC, and the latest learning-based methods from the past two years: IPDAE, PCT-PCC, Following [45, 46], we use point-to-point PSNR to measure the geometric accuracy and the number of bits per point to measure the compression ratio.

Implementation Our model is implemented using PyTorch [27] and CompressAI [4], trained on the NVIDIA 4090X GPU (24GB Memory) for 80,000 steps with a batch size of 48. We utilize the Adam optimizer [21] with an initial learning rate of 1e-4 and a decay factor of 0.5 every 30,000 steps, with β_1 set to 0.9 and β_2 set to 0.999. Since the positional encoding method requires the dimension (dim) to be a multiple of 6, we designed the bottleneck layer size to be 288. For diffusion, we employ a cosine preset noise parameter, setting the denoising steps T to 200, which is used for both training and testing.

Table 1: Objective comparison using BD-PSNR and BD-Rate metrics. G-PCC serves as the anchor. The best and second-best results are highlighted in **bold** and underlined, respectively.

Dataset	Metric	G-PCC	IPDAE	PCT-PCC	Diff-PCC
ShapeNet	BD-Rate (%)	-	-34.594	-87.563	**-99.999**
	BD-PSNR (dB)	-	+3.518	+8.651	**+11.906**
ModelNet10	BD-Rate (%)	-	-35.640	**-68.899**	-56.910
	BD-PSNR (dB)	-	+4.060	**+6.333**	+5.876
ModelNet40	BD-Rate (%)	-	-53.231	-34.127	**-56.451**
	BD-PSNR (dB)	-	+4.245	**+6.167**	+5.350
Avg.	BD-Rate (%)	-	-41.550	-63.530	**-71.117**
	BD-PSNR (dB)	-	+3.941	+4.384	**+7.711**
Time (s/frame)	Encoding	0.002	0.004	0.046	0.152
	Decoding	0.001	0.006	0.001	1.913

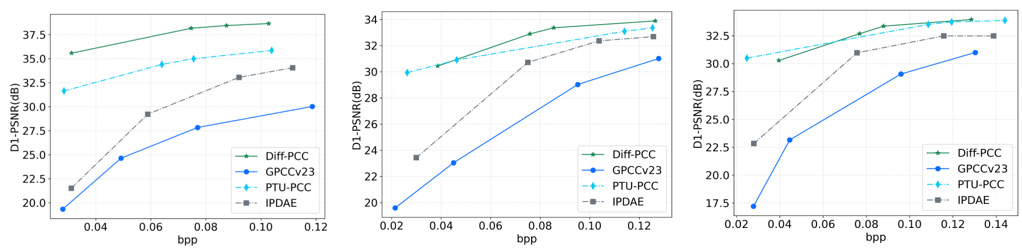

Figure 3: Rate-distortion curves for performance comparison. From left to right: ShapeNet, ModelNet10, and ModelNet40 dataset.

4.2 Baseline Comparisons

Objective Quality Comparison Table 1 shows the quantitative indicators using BD-Rate and BD-PSNR, and Fig. 3 demonstrates the rate-distortion curves of different methods. It can be seen that, under identical reconstruction quality conditions, our method achieves superior rate-distortion performance, conserving between 56% to 99% of the bitstream compared to G-PCC. At the most minimal bit rates, point ot point PSNR of our proposed method surpasses that of G-PCC by 7.711 dB.

Subjective Quality Comparison Fig 4 presents the ground truth and decoded point clouds from different methods. We choose three point cloud:airplane, chair ,and mug. to be tested across a comparable bits per pixel (bpp) range. The comparative analysis reveals that at the lowest code rate, our method preserves the ground truth's shape information to the greatest extent while simultaneously achieving the highest Peak Signal-to-Noise Ratio (PSNR).

4.3 Ablation Studies

We conduct ablation studies to examine the impact of key components in the model. Specifically, we investigate the effectiveness of low-frequency features, high-frequency features, and the loss function designed in Sec. 3.2.4. As shown in Table 2, utilizing solely low-frequency features to guide the reconstruction of the diffusion model results in a 20% reduction in the code rate, along with a decrease in the reconstruction quality by 0.397dB. This indicates that high-frequency features play an effective role in guiding the model during the reconstruction process. Conversely, discarding the low-frequency features, which represent the shape of the point cloud, leads to a reduction in the code rate and significantly diminishes the reconstruction quality. Therefore, we argue that the loss of the shape variable is not worth it. Lastly, we ascertain the impact of $\mathcal{D}_{CD}(x_0, \bar{x}_0)$, and the results indicate that this loss marginally increases the bits per point (bpp) while diminishing the reconstruction quality.

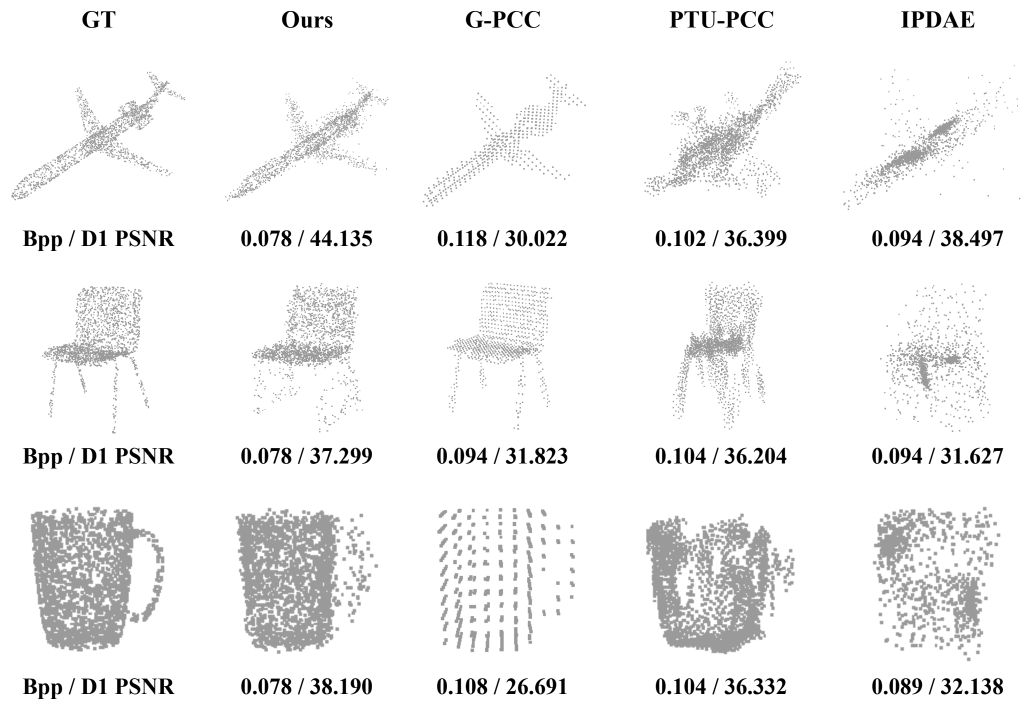

Figure 4: Subjective quality comparison. Example point clouds are selected from the ShapeNet dataset, each with 2k points.

Table 2: Ablation study of the proposed method. The original Diff-PCC serves as the anchor.

E_l backbone	E_h backbone	$\mathcal{D}_{CD}(x_0, \bar{x}_0)$	BD-PSNR (dB)	BD-Rate (%)
✔	✗	✔	-0.397	-20.637
✗	✔	✔	-2.276	-16.523
✔	✔	✗	-0.132	+4.658

5 Limitations

Although our method has achieved advanced rate distortion performance and excellent visual reconstruction results, there are several limitations that warrant discussion. Firstly, the encoding and decoding time are relatively long, which could potentially be improved by the acceleration techniques employed in several explorations [18, 19]. Secondly, the model is currently limited to compressing small-scale point clouds, and further research is required to enhance its capability to handle large-scale instances.

6 Conclusion

We propose a diffusion-based point cloud compression method, dubbed Diff-PCC, to leverage the expressive power of the diffusion model for generative and aesthetically superior decoding. We introduce a dual-space latent representation to enhance the representation ability of the conventional Gaussian priors in VAEs, enabling the Diff-PCC to extract expressive shape latents and facilitate the following diffusion-based decoding process. At the decoding side, an effective diffusion-based generator produces high-quality reconstructions by considering the shape latents as guidance to stochastically denoise the noisy point clouds. The proposed method achieves state-of-the-art compression performance while attaining superior subjective quality. Future works may include reducing the coding complexity and extending to large-scale point cloud instances.

References

[1] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

[3] Francesco Paolo Casale, Adrian Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[4] Ruihang Chu, Enze Xie, Shentong Mo, Zhenguo Li, Matthias Nießner, Chi-Wing Fu, and Jiaya Jia. Diffcomplete: Diffusion-based generative 3d shape completion. *Advances in Neural Information Processing Systems*, 36, 2024.

[5] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10850–10869, 2023.

[6] Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

[7] Kamak Ebadi, Lukas Bernreiter, Harel Biggie, Gavin Catt, Yun Chang, Arghya Chatterjee, Christopher E Denniston, Simon-Pierre Deschênes, Kyle Harlow, Shehryar Khattak, et al. Present and future of slam in extreme environments: The darpa subt challenge. *IEEE Transactions on Robotics*, 2023.

[8] Lili Fan, Junhao Wang, Yuanmeng Chang, Yuke Li, Yutong Wang, and Dongpu Cao. 4d mmwave radar for autonomous driving perception: a comprehensive survey. *IEEE Transactions on Intelligent Vehicles*, 2024.

[9] Chunyang Fu, Ge Li, Rui Song, Wei Gao, and Shan Liu. Octattention: Octree-based large-scale contexts model for point cloud compression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 625–633, 2022.

[10] Xiaoran Hao and Patrick Shafto. Coupled variational autoencoder. *arXiv preprint arXiv:2306.02565*, 2023.

[11] Yun He, Xinlin Ren, Danhang Tang, Yinda Zhang, Xiangyang Xue, and Yanwei Fu. Density-preserving deep point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2333–2342, 2022.

[12] Tianxin Huang, Jiangning Zhang, Jun Chen, Zhonggan Ding, Ying Tai, Zhenyu Zhang, Chengjie Wang, and Yong Liu. 3qnet: 3d point cloud geometry quantization compression network. *ACM Transactions on Graphics (TOG)*, 41(6):1–13, 2022.

[13] Zixuan Huang, Justin Johnson, Shoubhik Debnath, James M Rehg, and Chao-Yuan Wu. Pointinfinity: Resolution-invariant point diffusion models. *arXiv preprint arXiv:2404.03566*, 2024.

[14] Yiqi Jin, Ziyu Zhu, Tongda Xu, Yuhuan Lin, and Yan Wang. Ecm-opcc: Efficient context model for octree-based point cloud compression. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7985–7989, 2024.

[15] Weonyoung Joo, Wonsung Lee, Sungrae Park, and Il-Chul Moon. Dirichlet variational autoencoder. *Pattern Recognition*, 107:107514, 2020.

[16] M Krithika Alias AnbuDevi and K Suganthi. Review of semantic segmentation of medical images using modified architectures of unet. *Diagnostics*, 12(12):3064, 2022.

[17] Jin Sub Lee, Jisun Kim, and Philip M Kim. Score-based generative modeling for de novo protein design. *Nature Computational Science*, 3(5):382–392, 2023.

[18] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17535–17545, 2023.

[19] Qingguo Liu, Chenyi Zhuang, Pan Gao, and Jie Qin. Cdformer: When degradation prediction embraces diffusion model for blind image super-resolution. *arXiv preprint arXiv:2405.07648*, 2024.

[20] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[21] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *ArXiv*, abs/2112.03530, 2021.

[22] 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.

[23] Shentong Mo, Enze Xie, Ruihang Chu, Lanqing Hong, Matthias Niessner, and Zhenguo Li. Dit-3d: Exploring plain diffusion transformers for 3d shape generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[24] Dat Thanh Nguyen and André Kaup. Lossless point cloud geometry and attribute compression using a learned conditional probability model. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[25] Francesco Pezone, Osman Musa, Giuseppe Caire, and Sergio Barbarossa. Semantic-preserving image coding based on conditional diffusion models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13501–13505. IEEE, 2024.

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[28] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies.

[29] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. *PBG@ SIGGRAPH*, 3:111–121, 2006.

[30] Rui Song, Chunyang Fu, Shan Liu, and Ge Li. Efficient hierarchical entropy model for learned point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14368–14377, 2023.

[31] Rui Song, Chunyang Fu, Shan Liu, and Ge Li. Efficient hierarchical entropy model for learned point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14368–14377, 2023.

[32] Yu Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14453–14463, 2023.

[33] Lucas Theis, Tim Salimans, Matthew D Hoffman, and Fabian Mentzer. Lossy compression with gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022.

[34] Anwaar Ulhaq, Naveed Akhtar, and Ganna Pogrebna. Efficient diffusion models for vision: A survey. *arXiv preprint arXiv:2210.09292*, 2022.

[35] Juho-Pekka Virtanen, Sylvie Daniel, Tuomas Turppa, Lingli Zhu, Arttu Julin, Hannu Hyyppä, and Juha Hyyppä. Interactive dense point clouds in a game engine. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163:375–389, 2020.

[36] Jianqiang Wang, Dandan Ding, Zhu Li, and Zhan Ma. Multiscale point cloud geometry compression. In *2021 Data Compression Conference (DCC)*, pages 73–82. IEEE, 2021.

[37] Jianqiang Wang, Dandan Ding, and Zhan Ma. Lossless point cloud attribute compression using cross-scale, cross-group, and cross-color prediction. In *2023 Data Compression Conference (DCC)*, pages 228–237. IEEE, 2023.

[38] Jianqiang Wang and Zhan Ma. Sparse tensor-based point cloud attribute compression. In *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 59–64. IEEE, 2022.

[39] MPEG 3D Graphics WG 7 and Haptics Coding. G-pcc 2nd edition codec description. *ISO/IEC JTC 1/SC 29/WG 7*, 2023.

[40] MPEG 3D Graphics Coding WG 7. V-pcc codec description. *ISO/IEC JTC 1/SC 29/WG 7*, 2020.

[41] Yankun Wu, Yuta Nakashima, and Noa Garcia. Not only generative art: Stable diffusion for content-style disentanglement in art analysis. In *Proceedings of the 2023 ACM International conference on multimedia retrieval*, pages 199–208, 2023.

[42] Ruixiang Xue, Jiaxin Li, Tong Chen, Dandan Ding, Xun Cao, and Zhan Ma. Neri: Implicit neural representation of lidar point cloud using range image sequence. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8020–8024. IEEE, 2024.

[43] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. *arXiv*, 2019.

[44] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Kang You, Pan Gao, and Qing Li. Ipdae: Improved patch-based deep autoencoder for lossy point cloud geometry compression. In *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, pages 1–10, 2022.

[46] Kang You, Kai Liu, Li Yu, Pan Gao, and Dandan Ding. Pointsoup: High-performance and extremely low-decoding-latency learned geometry codec for large-scale point cloud scenes. *arXiv preprint arXiv:2404.13550*, 2024.

[47] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[48] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image diffusion model in generative ai: A survey. *arXiv preprint arXiv:2303.07909*, 2023.

[49] Junteng Zhang, Tong Chen, Dandan Ding, and Zhan Ma. Yoga: Yet another geometry-based point cloud compressor. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9070–9081, 2023.

[50] Junteng Zhang, Gexin Liu, Dandan Ding, and Zhan Ma. Transformer and upsampling-based point cloud compression. In *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, pages 33–39, 2022.

[51] Junteng Zhang, Jianqiang Wang, Dandan Ding, and Zhan Ma. Scalable point cloud attribute compression. *IEEE Transactions on Multimedia*, 2023.

[52] Junzhe Zhang, Tong Chen, Dandan Ding, and Zhan Ma. G-pcc++: Enhanced geometry-based point cloud compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1352–1363, 2023.

[53] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models.

[54] 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.

[55] Renrui Zhang, Liuhui Wang, Ziyu Guo, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Parameter is not all you need: Starting from non-parametric networks for 3d point cloud analysis. *arXiv preprint arXiv:2303.08134*, 2023.

[56] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.

[57] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5826–5835, October 2021.

[58] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1229, 2023.

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: Claims are clearly stated in abstract and introduction (Sec. 1). The experimental results (Sec. 4) match with these claims and reflect how much the results can be expected to generalize to other settings.

   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: Limitations are discussed in Sec. 5.

   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 include theoretical results.

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: The proposed architecture is fully described in Sec. 3, detailed instructions for replication is provided in the experimental setup section (Sec. 4.1).

   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: [No]

Justification: Source code will be made publicly available once paper is accepted.

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: Experimental setting and details are fully disclosed in Sec. 4.

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: Error bars are not reported due to the specificity of the compression task. The rate-distortion curve (Fig. 3) and Bjontegaard metric (Tab. 1) could be convincing enough.

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: Sufficient information on the computer resources is disclosed in the experiment setting (Sec. 4.1).

   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: This research does not involve human subjects or participants. This paper conform with the Code of Ethics in every respect.

   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: [NA]

    Justification: There is no societal impact of the work. The proposed method is limited to compression and reconstruction and cannot be used to generate deepfakes or disinformation.

    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: This paper poses no safeguard risks.

    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: This paper follows the license of the datasets used. Original papers are properly cited.

    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: This paper does not release new assets.

    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: This 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.
    - 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: This 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.