VVRec: Reconstruction Attacks on DL-based Volumetric Video Upstreaming via Latent Diffusion Model with Gamma Distribution

Rui Lu*, **Bihai Zhang***, **Dan Wang**

The Hong Kong Polytechnic University
{csrlu, csbzhang, csdwang}@comp.polyu.edu.hk

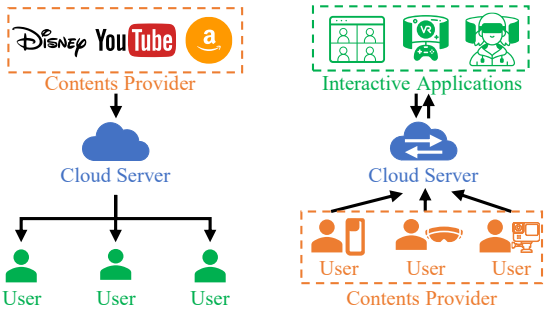
Abstract

With the popularity of 3D volumetric video applications, such as Autonomous Driving, Virtual Reality, and Mixed Reality, current developers have turned to deep learning for compressing volumetric video frames, i.e., point clouds for video upstreaming. The latest deep learning-based solutions offer higher efficiency, lower distortion, and better hardware support than traditional ones like MPEG and JPEG. However, privacy threats arise, especially *reconstruction attacks* targeting to recover the original input point cloud from the intermediate results. In this paper, we design **VVRec**, to the best of our knowledge, which is the first targeting DL-based **V**olumetric **V**ideo **Rec**onstruction attack scheme. VVRec demonstrates the ability to reconstruct high-quality point clouds from intercepted transmission intermediate results using four well-trained neural network modules we design. Leveraging the latest latent diffusion models with Gamma distribution and a refinement algorithm, VVRec excels in reconstruction quality, and color recovery, and surpasses existing defenses. We evaluate VVRec using three volumetric video datasets. The results demonstrate that VVRec achieves 64.70dB reconstruction accuracy, with an impressive 46.39% reduction of distortion over baselines.

1 Introduction

Currently, with the increasing broadcasting of 3D applications, customers can enjoy and experience the volumetric video technical, which are from fiction in the past, now coming to daily life using 3D commercial devices like Meta Oculus, Microsoft Hololens, etc. Emerging 3D applications in domains such as the Metaverse, Esports, and VR/MR require users to interactively share and display their surroundings with others. Unlike previous downstream applications where users only acted as video content receivers from cloud servers, these upstream applications, such as virtual meetings and VR games, necessitate active participation and real-time interaction. Every user is a unique content provider, and even a smartphone, e.g., the latest iPhone, can capture volumetric videos for Apple Vision Pro display and upstream them to others (Apple 2023).

Existing volumetric video is in the format of *point clouds* offering high precision and rich color representation (Zhang et al. 2023). Simultaneously, streaming techniques of volumetric video are developing rapidly (Pan et al. 2024) and public standards are introduced, e.g., G-PCC and V-PCC proposed by the ISO Moving Picture Experts Group (MPEG), and Google Draco and JPEG, etc. They apply a series of logical computations and contextual encoding based on octree structure features or 2D to 3D projection.



(a) Downstream Applications (b) Upstream Applications

Figure 1: Video applications where the content provider is evolving, from filming companies to individual users, can be categorized into Downstream and Upstream Applications.

Deep learning methods, known for their capability in context understanding and information encoding, have attracted attention, particularly in the volumetric video compression area (Luo et al. 2024; Gao, Ye et al. 2022). A DL-based approach involves constructing and training a pair of neural network (NN) models as an encoder and a decoder, compressing point clouds during NN inference (Quach, Valenzise, and Dufaux 2019). For instance, LVAC (Isik et al. 2021) utilizes a local coordinate-based model to input point clouds and compress them into intermediate results. These intermediate results are then transmitted to the cloud, where a corresponding decoder reverses them back to the original point clouds. In comparison to non-DL-based like G-PCC, V-PCC, DL-based methods offer advantages such as low distortion (Cao, Preda, and Zaharia 2019), excellent compression efficiency (Fang et al. 2022), and great hardware supports (Cui, Long et al. 2023). However, *privacy concerns* remain a significant threat during volumetric video serving (Lu et al. 2023). One of the attacks that can recover

*These authors contributed equally.

the original data from intermediate results is known as *Reconstruction Attacks*. It occurs during NN model inference across various machine-learning domains. For example, in language models, reconstruction attacks aim to reconstruct input text (Song and Raghunathan 2020; Brown et al. 2022), while in computer vision, they aim to reconstruct input images (Dosovitskiy and Brox 2016). However, no existing research has been done on activating the reconstruction attack on the upstreaming point cloud data.

The major challenge to attack DL-based point cloud compression models is the *distribution shift* between the training dataset used by the encoder-decoder pair of the service provider cloud and that used by attackers. This disparity prevents attackers from precisely training an identical decoder, resulting in significant reconstruction distortion. This disparity is attributed to the *distribution shift* in the training dataset between attackers and the encoder-decoder, which has been discussed widely in many machine learning domains, e.g., Transfer Learning (Zhuang et al. 2020), Federated Learning (Reisizadeh et al. 2020), etc. This shift worsens in point cloud data as it owns higher dimensional recording of the geometry information (Guo, Wang et al. 2021). Another challenge is that point clouds have a dynamic scale, often downsampled during compression by NN encoders for higher compression rates. Unlike images with specific resolutions, point clouds lacking precise scale may face significant degradation in reconstruction performance. The sampling or scale prediction is necessary for the attacker.

In our paper, we design **VVRec**, which is the first targeting DL-based **V**olumetric **V**ideo streaming **Rec**onstruction attack scheme. We design latent diffusion models, the state-of-the-art point cloud generative approach, with a Gamma distribution, to construct and train a reconstruction attacker capable of reconstructing the original point clouds in high quality. We also introduce a refinement algorithm, *PCR*, to further improve the quality of reconstructed point clouds.

To the best of our knowledge, VVRec is the first attempt to activate reconstruction attacks on DL-based volumetric video streaming. The contributions are as follows.

- We introduce VVRec, a reconstruction attack scheme based on SOTA latent diffusion models with four NN modules, to construct and train a reconstruction attacker capable of reconstructing the original point clouds in high quality.
- We develop a *PCR* algorithm to further improve the quality of the reconstructed point clouds.
- We compare VVRec on volumetric video datasets and attack seven victim models with other baselines. VVRec outperforms all in reconstruction accuracy.

2 Background and Related Work

2.1 Background

Volumetric Video Streaming. Existing volumetric video streaming methods can be categorized based on whether the transmission data constitutes the intermediate results of NN models or not (Gao, Ye et al. 2022): (1) non-DL-based compression, e.g., G-PCC and V-PCC by MPEG (MPEG 2020), Pleno (JPEG 2020) and Draco (Google 2022). (2) DL-based compression, e.g., PCGCv2 (Wang et al. 2021), SparsePCGC (Wang et al. 2023), etc.

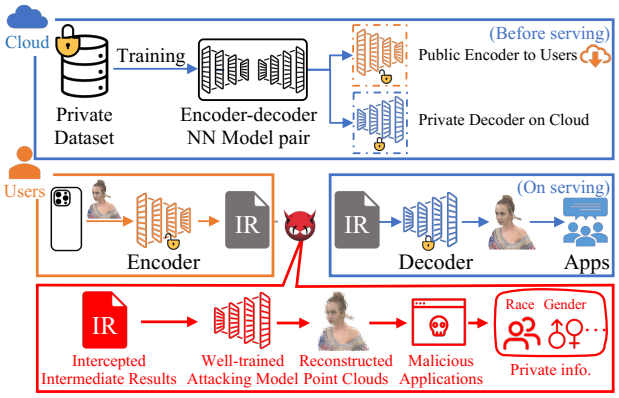


Figure 2: Workflow of existing DL-based volumetric video streaming and privacy risks.

A typical DL-based volumetric video compression workflow is shown in Fig. 2. As the 3D application service provider, the cloud constructs and trains a pair of NN models: an encoder and a decoder through a *private* dataset. Only the well-trained encoder will be distributed to users, while the decoder is secured on the cloud. Upon capturing volumetric video, users compress it into intermediate results using this encoder. These results are then transmitted to the cloud and decoded by the secured decoder, ultimately decompressing to the original volumetric video.

In this paper, we focus on recovering the volumetric video frame, i.e., point clouds, from the intercepted intermediate results during transmission in DL-based volumetric video compression. Some researchers aim to enhance non-DL-based approaches for improved efficiency (Akhtar et al. 2020), reduced distortion (Zhang et al. 2023), and privacy protection (Lu et al. 2023) using deep learning approaches. However, these methods are out of our scope because the transmitted data retains the same in non-DL-based approaches, which differs from the intermediate results of NN models in DL-based.

Attacks on Volumetric Video Streaming. Attack on streaming is a serious threat that involves unauthorized access and interception of streaming data. Attackers can manipulate or impair transmission data, potentially jeopardizing confidentiality and integrity (Ni, Zhang, and Vasilakos 2020). Attackers may adopt various tactics, including faking as a colluded or corrupted server to extract users' private information (Ren, Lv et al. 2023) or engaging in man-in-the-middle attacks (Vladimirov et al. 2022). Among such attackers, reconstruction attackers target to recover the original input point cloud from the intermediate results. It attracts significant attention in deep learning (Jegorova et al. 2022). For instance, in language models, reconstruction attacks aim to reconstruct input text (Brown et al. 2022), while in computer vision, they are to reconstruct input images (Lu et al. 2022).

Reconstruction attacks usually target the application layer, which is quite different from attackers cracking encryption in the transport layer. For example, a malicious cloud manipulates users into sending their video data, even if the transmission is secured through HTTPS. While encryption methods can safeguard the data during transmission, they do not prevent the exposure of private information contained within the video content. To illustrate this kind of attack, Fig. 2 provides an example in red, demonstrating the attacking process of reconstructing point clouds from intercepted intermediate results. Before starting an attack, the attacker needs to train a corresponding attacking model capable of reconstructing the intercepted intermediate results. The attack commences by hijacking the intermediate results transmitted from the user to the cloud. Reconstructed point clouds can be identified with sensitive information like facial features, gender, race, etc.

Various reconstruction attack approaches exist, including Supervised-based (Dosovitskiy and Brox 2016), GAN-based (Li et al. 2021), etc. However, the state-of-the-art generative solution, the latent diffusion model (Rombach et al. 2022), has not been used in reconstruction attacks yet. Another similar attack is the attribute inference attack (AIA) (Rigaki and Garcia 2023), which extracts attributes of original inputs, e.g., gender, race, etc, from intermediate results. AIA only needs to recover a few points with attribute features. Unlike AIA, reconstruction attacks demand more from attackers' capability as they need to recover the entire input data accurately.

2.2 Related Work

Diffusion models (DMs) have emerged as the most powerful generative models, showcasing remarkable performance in sample quality and good modality coverage (Huang, Lim, and Courville 2021), in area of image generation (Hu et al. 2024; Fan et al. 2023), audio synthesis (Kong et al. 2020), etc, compared to other generative schemes, e.g., GANs, VAEs, etc. DMs contain two phases: the *forward* process to arbitrary noises, e.g., Gaussian noise (Ho, Jain, and Abbeel 2020), Poisson noise (Xu et al. 2022), and the *reverse* process to approximate the target distribution by denoising. Despite the success across diverse tasks, DMs suffer from sampling delays for thousands of network inference iterations, making them costly in real-world applications. However, it makes them particularly suitable for our attack scenario, where strict time constraints and computation capability are not decisive requirements. Generally, the Gaussian distribution in DM may lead to poor sample generation, e.g., the prior hole problem (Sinha et al. 2021; Vahdat, Kreis, and Kautz 2021); however, *Denoising Diffusion Gamma Model* (DDGM) (Nachmani, Roman, and Wolf 2021) replaces it with the Gamma one, which has been proven to exhibit higher model capacity and improved convergence, particularly in high-dimensional data (Chang et al. 2023; Croitoru et al. 2023).

Latent diffusion models (LDMs) are initially introduced to enhance the efficiency of conventional DMs in high-resolution image generation (Rombach et al. 2022). These models encode high-dimensional input data into a low-dimensional latent space and train DMs within this latent space. Those latents from the high-dimensional point cloud are named *shape latent* (Zeng et al. 2022). Samples generated in the latent space are then decoded back to their original high-dimensional data. In reconstruction attacks, the intermediate results happen to be in a low dimension, i.e., *shape latents*, where the encoder of the victim model is the role of encoding high-dimensional to the low-dimensional latent space. All these features naturally contribute to the adoption of a new attacking scheme based on LDMs.

3 Design

3.1 Threat Model

Victim Model. It is assumed that the cloud server owns a large volumetric video dataset or a specific domain dataset, which is inaccessible to others. The cloud will use it to train a DL-based compression model pair, including an encoder and a decoder. The decoder is private and will never be publicly exposed. The encoder will be distributed in public. Users capture point clouds, compress them by the encoder, and then transmit them to the cloud for decoding.

Attack Model. Following existing reconstruction attacks (Rigaki and Garcia 2023), we assume the attacker owns sufficient computational resources and there is no latency constraint. Before the attack starts, the attacker accesses the encoder model in advance, possibly by spoofing attack. Although the attacker has volumetric data distinct from the cloud's, they can input their data into the encoder, generating intermediate results to train their attack model for recovering original inputs. During the attack, the attacker intercepts intermediate results from victim users and utilizes the trained attacker model to reconstruct the original point cloud. The success of the attack is determined by the reconstructed point clouds achieving a high similarity to the original ones.

3.2 Design Overview

We present the overview of VVRec in Fig. 3. The objective of a well-trained VVRec is to reconstruct the point cloud from the intercepted intermediate results, i.e., *shape latent*, in high quality. The source of the shape latent varies during training and inference. During training, the shape latent is derived from a dataset created by the attacker inputting their volumetric data into the compression encoder provided by the target victim model. In the attack, i.e., during the inference phase, the intercepted intermediate result is the shape latent that has been encoded by the victim user. In general, there are three major steps in VVRec:

- **Latent Points Generation** (§ 3.3): The shape latent is initially processed through the *Shape Latent Diffusion Model (SLDM)*, where it undergoes a forward diffusion process followed by a reverse denoising process to sample a new shape latent. Subsequently, the updated shape latent is transmitted to the *Latent Point Generator (LPG)* for encoding into a point cloud format, named *latent points*.
- **Point Cloud Generation** (§ 3.4): The latent points from LPG will first be sent to the *Latent Point Gamma Diffusion Model (LPGDM)* and that undergoes a forward diffusion

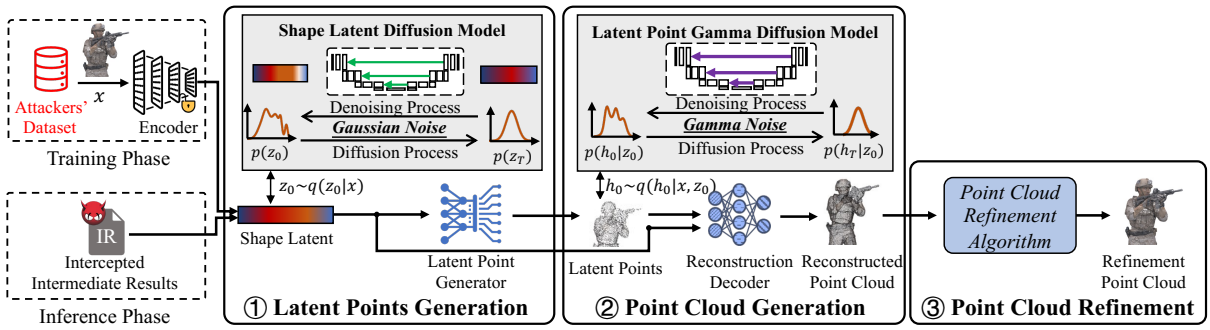


Figure 3: Workflow of VVRec activates a reconstruction attack, including training NN models in VVRec and reconstructing the given intercepted intermediate results to a point cloud.

process using a *Gamma* distribution and a reverse denoising process to sample a new latent point cloud. The resulting new latent point cloud, along with the shape latent, serves as the input to the *Reconstruction Decoder (RD)*, producing the *reconstructed point cloud*.

• **Point Cloud Refinement** (§ 3.5): To enhance the quality of the reconstructed point cloud generated by RD, a *Point Cloud Refinement (PRC) Algorithm* is designed to convert the point cloud into a mesh format, and resample to reconstruct a high-quality refinement point cloud.

3.3 Latent Points Generation

We first model the volumetric video frame, i.e., point cloud, defined as $\mathbf{x} \in \mathbb{R}^{(3+c)\times N}$, where N is the number of points with coordinates $xyz \in \mathbb{R}^3$. Here, c is the color representation, such as points are in RGB format if $c = 3$. $c = 0$ indicates that the volumetric video lacks color (or transmits color in the other way). The shape latent, denoted as $\mathbf{z}_0 \in \mathbb{R}^{D_\mathbf{z}}$ where $D_\mathbf{z}$ is the dimension of $\mathbf{z}_0$ determined by the victim model setup. Latent point is in point cloud format, denoted as $\mathbf{h}_0 \in \mathbb{R}^{(3+c+D_\mathbf{h})\times N}$, with N points with geometry coordinates $xyz \in \mathbb{R}^3$, and $D_\mathbf{h}$ is the additional latent feature dimension.

• ***Shape Latent Diffusion Model (SLDM).*** Shape latent $\mathbf{z}_0$ will be first sampled from the SLDM. Specifically, $\mathbf{z}_0$ follows a forward process to add Gaussian distribution noise in DMs.

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},\ t = 1, 2, ..., T, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{I}) \quad (1)$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. T denotes the number of diffusion steps and is set to T_{SL} in SLDM. The noise scheduling sequence $\{\beta_t\}$ is chosen such that the chain approximately converges to a standard Gaussian distribution (Ho, Jain, and Abbeel 2020), i.e., $\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{I})$, after T steps, if T is given large enough. The reverse process of SLDM is then defined as:

$$\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)), t = 1, 2, ..., T-1 \quad (2)$$

SLDM is designed to learn the reverse process with parameters $\boldsymbol{\theta}$ that inverts the forward diffusion by predicting $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$. The new shape latent $\mathbf{z}_0$ is sampled step by step from $\{\mathbf{z}_T, \mathbf{z}_{T-1}, ...\}$ by iteratively calculating Eq. 2. The simplified loss function for SLDM is defined as follows:

$$L_\mathbf{z}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}_t|\mathbf{x},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)||_2^2 \quad (3)$$

To support such functions, we follow (Zhou, Du et al. 2021) and build SLDM based on PVCNN (Liu et al. 2019). To model the distribution of shape latents, we add ResNet-like components into the SLDM with residual connections.

• ***Latent Point Generator (LPG).*** To construct latent points $\mathbf{h}_0$ in point cloud format from the sampled shape latent $\mathbf{z}_0$, we design LPG based on PVCNN model to support such functions with parameters ϕ as well. LPG takes $\mathbf{h}_0$ as input. To enable the conditioning on shape latents, we add adaptive Group Normalization modules to the original PVCNN model. Finally, $\mathbf{h}_0|\mathbf{z}_0$ is obtained with the dimension of $3+c+D_\mathbf{h}$. As LPG is intended to be trained in conjunction with the Reconstruction Decoder in the next section, we will defer the discussion of its loss function later.

3.4 Point Cloud Generation

In this section, the latent points $\mathbf{h}_0$, generated by LPG are initially processed by the LPGDM to sample new latent points. The resulting new latent points with the shape latent as conditions, serve as the input to the RD, producing the reconstructed point cloud $\mathbf{x}'$.

• ***Latent Point Gamma Diffusion Model (LPGDM).*** Similar to Eq. 1 SLDM, the sampling process of $\mathbf{h}_0$ starts with a forward process but incorporates Gamma distribution noise (Nachmani, Roman, and Wolf 2021) instead of Gaussian, as follows:

$$\mathbf{h}_t = \sqrt{\bar{\alpha}_t}\mathbf{h}_0 + (\bar{g}_t - \bar{k}_t\theta_t), t = 1, 2, ..., T-1 \quad (4)$$

where $\bar{g}_t \sim \Gamma(\bar{k}_t, \theta_t)$, $\theta_t = \sqrt{\bar{\alpha}_t}\theta_0$, $\bar{k}_t = \sum_{i=1}^{t} \frac{\beta_t}{\alpha_t\theta_0^2}$. θ_0 is the initial Gamma distribution scale hyperparameter. The reverse process is then defined as follows :

$$\begin{aligned}\mathbf{h}_{t-1} = &\frac{1}{\sqrt{\alpha_t}}(\mathbf{h}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\psi}}(\mathbf{h}_t, \mathbf{z}_0, t)) \\ &+ \beta_t\frac{\gamma_{t-1} - \theta_{t-1}\bar{k}_{t-1}}{\sqrt{1-\bar{\alpha}_t}},\ t = 1, 2, ..., T-1\end{aligned} \quad (5)$$

where $\gamma_t \sim \Gamma(\theta_t.\bar{k}_t)$, and $T := T_{LPG}$ is the diffusion steps number.

Algorithm 1: Point Cloud Refinement Algorithm

Input: Reconstructed Point Cloud $\mathbf{x}' \in \mathbb{R}^{3+c+N}$, Maximum #points N_{max}, Maximal quality gradient δ_{max}

Output: Refinement Point Cloud $\mathbf{x}'' \in \mathbb{R}^{3+c+N'}$

1 $M \leftarrow SAP(\mathbf{x}')$;
2 **while** $\#\mathbf{y} \leq N_{max}$ **do**
3 $\mathbf{y} \leftarrow PDS(M, d_0)$;
4 **if** $p2plane\_PSNR(\mathbf{y}, \mathbf{y_{last}}) \leq \delta_{max}$ **then**
5 Increase upsampling precision d_0 ;
6 $\mathbf{y_{last}} \leftarrow \mathbf{y}$;
7 **else**
8 **Break** ;
9 Output $\mathbf{x}''$.

LPGDM is designed to learn the reverse process with parameters ψ by predicting $\boldsymbol{\epsilon}_{\psi}(\mathbf{h}_t|\mathbf{z}_0, t)$. LPGDM applies PVCNN structure to efficiently model the reverse process.

The latent point clouds $\mathbf{h}_0|\mathbf{z}_0$ are iteratively sampled by stepwise calculations according to Eq. 5. The loss function of LPGDM is defined as follows:

$$L_{\mathbf{h}}(\psi) = \mathbb{E}_{\mathbf{h}_t|\mathbf{z}_0, \bar{g}_t \sim \Gamma(\bar{k}_t, \theta_t)} || \frac{(\bar{g}_t - \bar{k}_t\theta_t)}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\epsilon}_{\psi}(\mathbf{h}_t, t)||_2^2 \tag{6}$$

• ***Reconstruction Decoder (RD).*** Following the sampling of latent points $\mathbf{h}_0|\mathbf{z}_0$ by LPGDM, a reconstruction decoder model with parameters $\boldsymbol{\xi}$ is devised to reconstruct the point cloud $\mathbf{x}'$ using the input latent points and shape latent $\mathbf{z}_0$. The design of RD is constructed as a VAE based on PVCNN. Different from the LPG, the output of RD conditions on two inputs, $\mathbf{z}_0$ and $\mathbf{h}_0$. To accommodate the conditioning on two latent encodings, we replace the original group normalization in PVCNN with adaptive group normalization (AdaGN). To be specific, the shape latent $\mathbf{z}_0$ is cut into factor and bias, and the latent point $\mathbf{h}_0$ is group-normalized. Finally, the product of group-normalized $\mathbf{h}_0$ and factor plus the bias is the output of the AdaGN layer. The reconstructed point cloud $\mathbf{x}'$ is the weighted summation of RD output and latent points. The training of RD is conducted in conjunction with LPG. The loss function is designed based on the maximization of a modified variational lower bound on the data log-likelihood (ELBO) (Hoffman and Johnson 2016):

$$\begin{aligned} L_{\text{ELBO}}(\boldsymbol{\phi}, \boldsymbol{\xi}) = \mathbb{E}_{\mathbf{x}', \mathbf{z}_0, \mathbf{h}_0|\mathbf{z}_0}[\log p_{\boldsymbol{\xi}}(\mathbf{x}'|\mathbf{h}_0, \mathbf{z}_0) - \\ \lambda_{\mathbf{z}} D_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}_0)|p(\mathbf{z}_0)) - \lambda_{\mathbf{h}} D_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{h}_0|\mathbf{z}_0)|p(\mathbf{h}_0))]. \end{aligned} \tag{7}$$

where $p(\mathbf{h}_0), p(\mathbf{z}_0) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $p_{\boldsymbol{\xi}}(\mathbf{x}'|\mathbf{h}_0, \mathbf{z}_0)$ denotes RD. Here, $\lambda_{\mathbf{z}}$ and $\lambda_{\mathbf{h}}$ are hyperparameters balance reconstruction accuracy and K-L regularization D_{KL} (Zeng et al. 2022).

3.5 Point Cloud Refinement

To generate a point cloud of high quality, we develop a Point Cloud Refinement Algorithm (PCR) as shown in Algorithm 1 to refine the reconstructed point cloud. It is able to smooth the point cloud surface and simultaneously cancel out the noisy points from the LPGDM. In Line 1, We initially utilize a well-trained *Shape As Points (SAP)* (Peng et al. 2021), which efficiently extracts high-quality watertight meshes from point clouds based on Differentiable Poisson Surface Reconstruction algorithm. It establishes relationships among independent points in $\mathbf{x}'$ to mesh, denoted M. From Lines 2 to 8, we iteratively generate point clouds by increasing sampling precision d_0 via Poisson Disk Sampling (PDS) (Bowers et al. 2010) and calculating the similarity (p2plane-PSNR) between the sampled point clouds and the last generated one. We will yield a refinement point cloud, denoted as $\mathbf{x}''$ with N' point quantity until the similarity is higher than the maximal quality gradient δ_{max} or exceeding the maximal point number N_{max}.

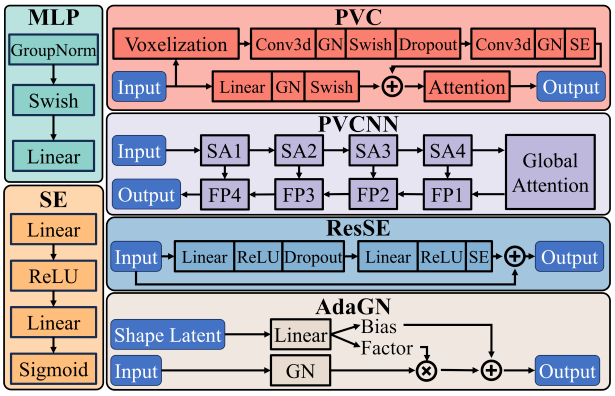


Figure 4: Basic blocks of VVRec.

4 Implementation and Evaluation

4.1 Implementation

We establish VVRec by building the two diffusion models, SLDM and LPDGM, plus an LPG and an RD on PVCNN (Liu et al. 2019) for the point cloud. Point cloud refinement is achieved through the Shape-as-Points (SAP) (Peng et al. 2021), which generates a mesh from the reconstructed point cloud. Utilizing Alg. 1, we produce a refined point cloud. The training of VVRec and the two diffusion models are developed based on HuggingFace. Victim models are supported by OpenPointCloud project (Gao, Ye et al. 2022).

Training details. There are two stages of training VVRec. Firstly, we train the LPG with RD by optimizing the loss function in Eq. 7. During this process, the latent point $\mathbf{h}_0$ is generated from the LPG model. The reconstructed point cloud $\mathbf{x}'$ is sampled from its posterior distribution conditioning on the shape latent $\mathbf{z}_0$ and the latent point $\mathbf{h}_0$ estimated by the RD. Secondly, we freeze the LPG and RD models to train SLDM and LPGDM on the shape latent $\mathbf{z}_0$ and latent point $\mathbf{h}_0$, minimizing the loss functions defined in Eq. 3 and 6 respectively. The number of steps, T_{SL}, T_{LPG} in SLDM and LPGDM are set to 1000. We utilize Adam optimizer with the learning rate of LPG/RD, SLDM, and LPGDM as 10e-3, 10e-4, and 10e-4, respectively. The value of N is configured as 2e4. The number of parameters is 174M in total, and the computation complexity is 1.05 TFLOPS. The overall training time in a 32 batch size is 500 hours.

Input: latent points h_t at t, shape latent z_0 of dimension $(1 \times D_z)$
Output: new shape latent z_0 of dimension $(1 \times D_z)$
Time embedding layer:
Sinusoidal embedding dimension = 128
Linear (128, 512)
LeakyReLU (0.1)
Linear (2048)
Linear (128, 2048)
Addition (linear output, time embedding)
ResSE (2048, 2048) x 8
Linear (2048, 128)

Table 1: Architecture of SLDM.

Details of model architecture. The blocks used for implementing modules of VVRec are displayed in Fig. 4. Here, multi-layer perceptron (MLP), point-voxel convolution (PVC), set abstraction (SA), and feature propagation (FP) are the four building blocks of our PVCNN-like modules. For the details of SA and FP modules, please refer to PointNet++(Qi et al. 2017). The squeeze-and-excitation (SE) module is inherited from (Hu et al. 2018), and the ResSE module denotes the ResNet-like residual module with the SE layer. The adaptive group normalization (AdaGAN) layer is designed for conditioning the shape latent. The details of VAE models and diffusion models are displayed as follows:

• **VAE Models** Our latent point generator (LPG) and reconstruction decoder (RD) are constructed as VAEs. Both LPG and RD consist of PVC layers, the global attention layer, MLP layers, the dropout layer, and the linear output layer. All dropout layers in the VAEs are implemented with a dropout probability of 0.1. To fulfill conditioning on the shape latents, we replace them with the AdaGN layers. The initial weight scale is 0.1. We also add residual connections in various layers. In the LPG, the raw input point cloud plus the product of 0.01 and the mean of the predicted latent point form the final output. In the RD, the final output is the summation of the sampled latent point cloud and 0.01 times the predicted point cloud.

• **Diffusion Models** We illustrate the details of the Shape Latent Diffusion Model (SLDM) and Latent Point Gamma Diffusion Model (LPGDM) in Tab. 1 and Tab. 2, respectively. Similar to the VAE models, a dropout probability of 0.1 is adopted for all dropout layers in the SLDM and LPGDM, and we replace all GN layers with AdaGN layers to realize conditioning on the shape latent. For the AdaGN layers, the initial weight scale is 0.1. For all SA and FP layers, the point features are concatenated with time embeddings to achieve the diffusion-denoising process.

4.2 Experimental Setup

Testbed. Our experiments adopt a powerful workstation featuring dual NVidia RTX 4090 GPUs with a total graphic memory of 48GB, along with an Intel i9 CPU.

Datasets. Four datasets are adopted, which are: 1) *8i Voxelized Full Bodies (8iv2)* (Krivokuca, Chou et al. 2018), 2) *Owlii Dynamic Human Textured (Owlii)* (Yi, Yao, and Ziyu 2017), 3) *Microsoft Voxelized Upper Bodies (MVUB)* (Charles, Qin et al. 2012), 4) *UVG-VPC* (Gautier et al. 2023). All of them are certified by the JPEG as the benchmark dataset for volumetric video streaming.

Input: latent points h_t at t, shape latent z_0				
Output: new latent points h_0				
Time embedding layer				
Sinusoidal embedding dimension = 64				
Linear (64, 64)				
LeakyReLU (0.1)				
Linear (64, 64)				
	SA1	SA2	SA3	SA4
# PVC Layers	2	1	1	N/A
# PVC hidden dimension	32	64	128	N/A
# PVC voxel grid size	32	16	8	N/A
# Grouper center	1024	256	64	16
# Grouper radius	0.1	0.2	0.4	0.8
# Grouper neighbors	32	32	32	32
# MLP layers	2	2	2	3
# MLP output dimension	32,32	64,128	128,128	128,128,128
# Attention dimension	N/A	128	N/A	N/A
Global attention, hidden dimension of 256				
	FP1	FP2	FP3	FP4
# MLP layers	2	2	2	3
# MLP output dimension	128,128	128,128	128,128	128,128,64
# PVC layers	3	3	2	2
# PVC hidden dimension	128	128	128	64
# PVC voxel grid size	8	8	16	32
MLP: (64,128)				
Dropout				
Linear: (128, $3 + c + D_h$)				

Table 2: Architecture of LPGDM.

Victim models. We use five DL-based volumetric video streaming models as our attacking targets, including 1) PCGCv1 (Wang, Zhu et al. 2021), 2) PCGCv2 (Wang et al. 2021), 3) SparsePCGC (Wang et al. 2023), 4) Pcc-geo-cnn-v1 (Quach, Valenzise, and Dufaux 2019), 5) Pcc-geo-cnn-v2 (Quach, Valenzise, and Dufaux 2020), which are lossy models. 6) VoxelDNN (Nguyen and Kaup 2022) and 7) MSVoxelDNN (Nguyen et al. 2021) are lossless models. Without further specification, the bit rate of these victim models is set to 0.4 bpp. The evaluation results of VoxelDNN and MSVoxelDNN, and other bit rates are listed in the Appendix.

Baselines. We compare VVRec performance with four reconstruction attack methods and the decoder coupled with the victim encoder: • ***Supervised*** has symmetric NN structure to the victim encoder model. The input point clouds of the victim encoder are the training labels and the output intermediate results are the training data. • ***SP-GAN (Li et al. 2021)*** trained a generator and a discriminator where the generator input intermediate results from the victim encoder and output reconstructed point cloud and adversarial training with the discriminator. • ***VG-VAE (Anvekar et al. 2022)*** applies a hierarchical VAE. • ***PD-Flow (Mao et al. 2022)*** is a point cloud denoising framework with normalizing flows. • ***Optimal*** is the decoder coupled with the victim encoder and serves as the upper bound of reconstruction performance.

Attacker Dataset	Victim Dataset	Victim Encoder	*PCGCv1*		*PCGCv2*		*SparsePCGC*		*Pcc-geo-cnn-v1*		*Pcc-geo-cnn-v2*	
		Metrics	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2
MVUB	8iv2	① *Optimal*	72.02	76.85	73.75	77.61	76.88	80.97	68.16	72.07	70.78	74.39
		② *Supervised*	45.62	50.01	41.03	45.75	40.26	45.06	46.81	51.36	41.57	46.01
		③ *SP-GAN*	50.15	54.95	45.71	50.04	44.36	48.91	48.15	52.93	45.57	50.81
		④ *VG-VAE*	48.05	52.78	43.61	47.84	42.16	46.51	46.02	50.73	43.67	48.95
		⑤ *PD-Flow*	47.25	51.48	41.71	46.92	41.33	45.92	44.68	49.03	42.66	47.80
		⑥ *VVRec*	**58.75**	**63.52**	**56.21**	**60.82**	**53.72**	**58.01**	**59.24**	**63.78**	**57.01**	**62.08**
MVUB	Owlii	*Optimal*	72.18	76.45	73.48	77.56	76.78	80.57	67.85	72.12	70.45	74.03
		Supervised	45.68	50.02	41.07	45.73	40.29	45.11	46.88	51.29	41.58	46.07
		SP-GAN	50.06	54.98	45.70	50.09	44.28	48.98	48.13	52.92	45.61	50.84
		VG-VAE	48.09	52.77	43.59	47.86	42.17	46.53	46.06	50.74	43.66	48.92
		PD-Flow	47.24	51.45	41.73	46.96	41.23	46.01	44.72	49.00	42.69	47.77
		VVRec	**58.72**	**63.49**	**56.22**	**60.80**	**53.75**	**58.04**	**59.27**	**63.75**	**57.02**	**62.15**
MVUB	UVG-VPC	*Optimal*	73.08	77.45	74.56	78.21	77.36	81.73	69.12	73.01	71.45	74.98
		Supervised	45.01	49.12	40.48	45.06	39.72	44.29	45.93	50.88	41.02	45.62
		SP-GAN	49.45	54.16	44.99	49.78	43.89	48.20	47.45	52.33	45.01	50.14
		VG-VAE	47.56	52.08	43.02	47.17	41.65	45.97	45.48	50.09	43.04	48.17
		PD-Flow	46.91	50.89	41.04	46.13	40.56	45.14	43.98	48.73	42.05	47.17
		VVRec	**58.05**	**62.89**	**55.54**	**60.08**	**53.16**	**57.48**	**58.78**	**63.26**	**56.25**	**61.67**
Owlii	8iv2	*Optimal*	72.02	76.85	73.75	77.61	76.88	80.97	68.16	72.07	70.78	74.39
		Supervised	50.03	54.49	45.79	50.46	44.31	49.20	49.11	53.64	46.66	51.37
		SP-GAN	52.18	56.36	48.86	53.72	47.77	52.57	51.21	55.67	48.77	53.67
		VG-VAE	50.16	54.38	46.88	51.71	45.76	50.47	49.03	53.36	46.97	51.87
		PD-FLow	49.19	53.06	44.86	50.63	44.57	49.48	48.20	52.47	44.97	49.86
		VVRec	**60.28**	**64.70**	**58.01**	**62.43**	**55.88**	**60.97**	**59.97**	**64.70**	**58.81**	**63.27**
UVG-VPC	8iv2	*Optimal*	72.02	76.85	73.75	77.61	76.88	80.97	68.16	72.07	70.78	74.39
		Supervised	49.93	54.27	45.82	50.63	44.01	49.14	48.73	53.03	46.45	51.27
		SP-GAN	52.33	56.04	47.99	53.02	48.07	52.65	51.01	55.68	48.72	53.69
		VG-VAE	50.03	54.16	46.67	51.73	44.98	50.26	51.31	55.89	46.98	51.89
		PD-FLow	49.02	53.14	44.97	50.69	44.29	49.37	48.01	52.38	45.08	49.98
		VVRec	**60.33**	**64.85**	**57.89**	**62.17**	**55.69**	**60.87**	**60.32**	**65.27**	**58.64**	**63.43**

Table 3: Reconstruction quality results measured by p2p-PSNR (M1↑) and p2plane-PSNR (M2↑) with bitrate 0.4bpp.

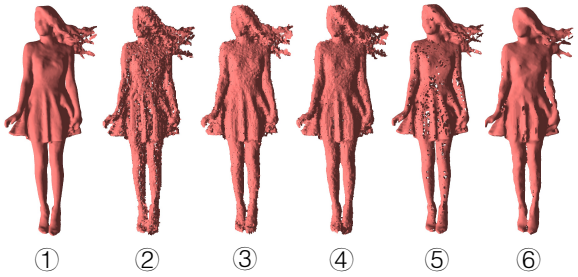


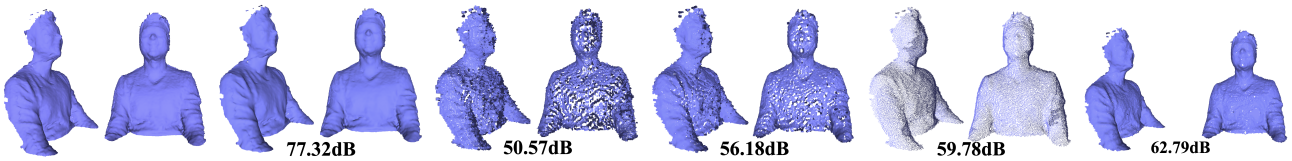
Figure 5: Visualization of reconstruction result of VVRec and baselines on trace *redandblack@8i* during attacking victim encoder PCGCv1 on above Table 1.

Evaluation metrics. We use the point-to-point Peak Signal-to-Noise Ratio (p2p-PSNR) and point-to-plane PSNR (p2plane-PSNR) (Tian et al. 2017), denoted as M1 and M2, respectively. The higher value represents higher similarity and better reconstruction performance.

4.3 Reconstruction Performance

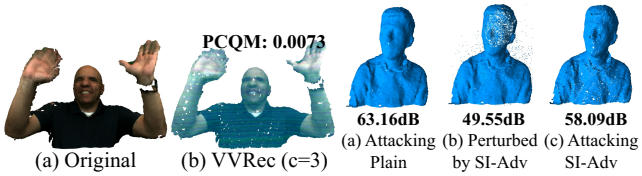
The performance of VVRec is measured by M1 and M2 between the reconstructed point cloud and the original one. In this context, the attacker's dataset is used for training the attacker model, while the testing dataset is the dataset encoded by the victim encoder.

As shown in Tab. 3, when the attacker dataset is MVUB and the victim dataset is 8iv2, VVRec exhibits only a marginal decline at about 18.91% in M1, averaging on seven victim models compared to the Optimal, representing minimal quality degradation among all attacking schemes. The supervised method records the lowest M1 and M2, likely due to its training data being utterly different from the victim model. Additionally, VVRec outperforms SP-GAN with an improvement of M1 up to 33.18%. The results show that the latent diffusion scheme in VVRec can converge to the distribution of the decoder model, leading to high reconstruction performance. Notably, SP-GAN shows varying performance under different victim models, possibly attributed to challenges in convergence and local optimum traps. We observe consistent results for other combinations of the attacker and victim datasets. VVRec exhibits up to 37.20% improvement in M1 compared to other methods, with only 12.02% to 31.28% lower performance than the Optimal. Moreover, compared to the case when the attacker dataset is MVUB, the reconstruction performance of all attack methods is enhanced when trained on Owlii and UVG-VPC. This



(a) Original (b) Paired decoder (c) VVRec-A (d) VVRec-B (e) VVRec-C (f) VVRec

Figure 6: Visualization of ablation study on breakdown version of VVRec, measured by p2plane-PSNR.



(a) Original (b) VVRec (c=3) (a) Attacking Plain (b) Perturbed by SI-Adv (c) Attacking SI-Adv

Figure 7: Study of color recovery on *Ricardo@MVUB*.

Figure 8: Study under protection on *Andrew@MVUB*.

improvement can be attributed to the similarity in volumetric video content shared by Owlii, UVG-VPC and 8iv2, all featuring full bodies, while MVUB represents half-upper body content. The improvement is more apparent when the attacker training dataset is a subset of 8iv2. Fig. 5 displays the visual comparison between VVRec and baselines vividly.

4.4 Ablation Study

To explore the virtue of each component in VVRec, we utilize three breakdown versions of VVRec, *VVRec-A*, *VVRec-B*, and *VVRec-C*. *VVRec-A* does not have SLDM and LPDGM; only LPG and RD are implemented. *VVRec-B* replaces the Gamma distribution in LPDGM with the Gaussian. *VVRec-C* does not apply refinement in VVRec.

We use M2 as the assessment metric. We train attacker models on MVUB and attack the intermediate results generated by PCGCv2 trained on 8iv2. In Fig. 6, VVRec-A exhibits inferior performance compared to VVRec, achieving the lowest reconstruction quality due to the absence of the diffusion model scheme, making VVRec-C theoretically a supervised method suffering from serious data distribution shift. VVRec-B lags behind VVRec by 6.61dB, indicating that the Gaussian distribution in high-dimensional latent points performs worse than the Gamma distribution in the latent diffusion model. VVRec-C, lacking a refinement module, experiences a 4.79% reduction in quality compared to VVRec, underscoring the significance of the refinement module in further enhancing reconstruction performance.

4.5 Study On Color Recovery

Here, we try to investigate VVRec's capability of color recovery. As mentioned in Section 3.3, VVRec supports color representations when appropriate c values are assigned. We set LVAC (Isik et al. 2021) as the victim model as it can encode the color representation within intermediate results. We use PCQM (Meynet et al. 2020) to evaluate those point clouds with color, and the visualization of attacking results of trace *Ricardo* from MVUB and the one from victim decoder is shown in Fig. 7. Compared to the Optimal, the overall average PCQM degradation is only 26.13%, indicating that VVRec can reconstruct both the geometry and color information of point clouds in high quality.

4.6 Study On Possible Protection

Since no existing method is specifically designed for protecting against point cloud reconstruction attacks, we explore a typical protection solution, SI-Adv (Huang et al. 2022), designed for attribute inference attacks protection. It introduces adversarial perturbations by injecting a few points into the point clouds to confuse the analytics results. For instance, it adds noise points on the face to disturb the gender classification. The visualization results are displayed in Fig. 8. The overall average reconstruction accuracy, measured by p2plane-PSNR, indicates that this protection approach has only a minimal impact on reconstruction performance, with less than 10% degradation.

5 Limitations

Regarding limitations, VVRec is designed specifically for DL-based volumetric upstreaming, where privacy is a concern during transmission. It does not apply to downstream applications, as their content is meant for public dissemination. Secondly, we assume that the attacker has no latency constraints, as even recovering a single frame could result in significant privacy breaches. However, this assumption may be limiting in scenarios that require rapid response due to sampling delays. Third, as protection solutions for DL-based streaming continue to evolve, our attack schemes could potentially be countered by emerging defensive approaches.

6 Conclusions

In conclusion, this paper presents VVRec, a novel reconstruction attack scheme targeting DL-based volumetric video upstreaming. VVRec demonstrates the ability to reconstruct high-quality point clouds from intercepted intermediate results using four well-trained neural network models: two latent diffusion models (SLDM, LPGDM), a generator (LPG), and a decoder (RD). Leveraging the latest latent diffusion models with Gamma distribution and a refinement module, VVRec excels in reconstruction quality and color recovery and surpasses existing defenses. The significance of our work extends to understanding the privacy threats in volumetric video applications where safeguarding sensitive information during streaming is crucial.

Acknowledgments

Dan Wang's work is supported by RGC GRF 15200321, 15201322, 15230624, RGC-CRF C5018-20G, ITC ITF-ITS/056/22MX, and PolyU 1-CDKK, G-SAC8.

References

Akhtar, A.; et al. 2020. Point cloud geometry prediction across spatial scale using deep learning. In *Proc. of IEEE Visual Communications and Image Processing (VCIP'20)*.

Anvekar, T.; Tabib, R. A.; Hegde, D.; and Mudengudi, U. 2022. VG-VAE: A Venatus Geometry Point-Cloud Variational Auto-Encoder. In *Proc. of the 35th IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)'22*. New Orleans, LA, USA.

Apple. 2023. Apple introduces spatial video capture on iPhone 15 Pro.

Bowers, J.; Wang, R.; Wei, L.-Y.; and Maletz, D. 2010. Parallel Poisson disk sampling with spectrum analysis on surfaces. *ACM Transactions on Graphics (TOG)*, 29(6): 1–10.

Brown, H.; Lee, K.; Mireshghallah, F.; Shokri, R.; et al. 2022. What does it mean for a language model to preserve privacy? In *Proc. of ACM Conference on Fairness, Accountability, and Transparency (FAccT'22)*. Seoul, South Korea.

Cao, C.; Preda, M.; and Zaharia, T. 2019. 3D Point Cloud Compression: A Survey. In *Proc. of ACM International Conference on 3D Web Technology (Web3D'19)*. LA, USA.

Chang, Z.; et al. 2023. On the Design Fundamentals of Diffusion Models: A Survey. *arXiv:2306.04542*.

Charles, L.; Qin, C.; et al. 2012. Microsoft Voxelized Upper Bodies - A Voxelized Point Cloud Dataset.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Cui, M.; Long, J.; et al. 2023. OctFormer: Efficient octree-based transformer for point cloud compression with local enhancement. In *Proc. of AAAI Conference on Artificial Intelligence, (AAAI'23)*. Washington, DC, USA.

Dosovitskiy, A.; and Brox, T. 2016. Inverting visual representations with convolutional networks. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'16)*. Las Vegas, NV, US.

Fan, W.-C.; Chen, Y.-C.; Chen, D.; Cheng, Y.; et al. 2023. Frido: Feature pyramid diffusion for complex scene image synthesis. In *Proceedings of conference on artificial intelligence, (AAAI'23)*. Washington, USA.

Fang, G.; Hu, Q.; Wang, H.; Xu, Y.; and Guo, Y. 2022. 3DAC: Learning Attribute Compression for Point Clouds. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. New Orleans, LA, USA.

Gao, W.; Ye, H.; et al. 2022. OpenPointCloud: An Open-Source Algorithm Library of Deep Learning Based Point Cloud Compression. In *Proc. of ACM international conference on multimedia (MM'22)*. Lisbon, Portugal.

Gautier, G.; et al. 2023. UVG-VPC: Voxelized Point Cloud Dataset for Visual Volumetric Video-based Coding. In *Proc. of 15th International Conference on Quality of Multimedia Experience (QoMEX'23)*. Ghent, Belgium.

Google. 2022. Draco 3D Graphics Compression.

Guo, Y.; Wang, H.; et al. 2021. Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12): 4338–4364.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. arXiv:2006.11239.

Hoffman, M. D.; and Johnson, M. J. 2016. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference*.

Hu, J.; et al. 2018. Squeeze-and-Excitation Networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. Salt Lake City, UT, USA.

Hu, T.; Zhang, J.; Yi, R.; et al. 2024. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence, (AAAI'24)*. Vancouver, BC, Canada.

Huang, C.-W.; Lim, J. H.; and Courville, A. 2021. A Variational Perspective on Diffusion-Based Generative Models and Score Matching. arXiv:2106.02808.

Huang, Q.; Dong, X.; Chen, D.; Zhou, H.; Zhang, W.; and Yu, N. 2022. Shape-invariant 3d adversarial point clouds. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. New Orleans, LA, US.

Isik, B.; et al. 2021. LVAC: Learned Volumetric Attribute Compression for Point Clouds using Coordinate Based Networks. *arXiv preprint:2111.08988*.

Jegorova, M.; Kaul, C.; Mayor, C.; O'Neil, A. Q.; Weir, A.; Murray-Smith, R.; and Tsaftaris, S. A. 2022. Survey: Leakage and privacy at inference time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

JPEG. 2020. *Final Call for Evidence on JPEG Pleno Point Cloud Coding*.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint:2009.09761*.

Krivokuca, M.; Chou, P. A.; et al. 2018. 8i voxelized surface light field dataset.

Li, R.; Li, X.; Hui, K.-H.; et al. 2021. SP-GAN:Sphere-Guided 3D Shape Generation and Manipulation. *ACM Transactions on Graphics*, 40(4).

Liu, Z.; et al. 2019. Point-Voxel CNN for Efficient 3D Deep Learning. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS'19)*. Vancouver, Canada.

Lu, R.; Shi, S.; Wang, D.; Hu, C.; and Zhang, B. 2022. Preva: Protecting Inference Privacy through Policy-based Video-frame Transformation. In *Proc. of IEEE/ACM Symposium on Edge Computing (SEC'22)*. Seattle, WA, USA.

Lu, R.; Wei, L.; Zhu, S.; Hu, C.; and Wang, D. 2023. Pagoda: Privacy Protection for Volumetric Video Streaming through Poisson Diffusion Model. In *Proc. of ACM International Conference on Multimedia (MM'23)*. Ottawa, Canada.

Luo, A.; Song, L.; Nonaka, K.; Unno, K.; Sun, H.; Goto, M.; and Katto, J. 2024. SCP: Spherical-Coordinate-Based Learned Point Cloud Compression. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence, (AAAI'24)*. Vancouver, BC, Canada.

Mao, A.; Du, Z.; Wen, Y.-H.; Xuan, J.; and Liu, Y.-J. 2022. PD-Flow: A Point Cloud Denoising Framework with Normalizing Flows. In *Proceedings of the 17th European Conference on Computer Vision (ECCV'22)*. Tel Aviv, Israel.

Meynet, G.; Nehmé, Y.; Digne, J.; et al. 2020. PCQM: A full-reference quality metric for colored 3D point clouds. In *Proc. of IEEE International Conference on Quality of Multimedia Experience (QoMEX'20)*. Athlone, Ireland.

MPEG. 2020. GitHub - MPEGGroup/mpeg-pcc-tmc2: Video codec based point cloud compression.

Nachmani, E.; Roman, R. S.; and Wolf, L. 2021. Denoising diffusion gamma models. *arXiv preprint:2110.05948*.

Nguyen, D. T.; and Kaup, A. 2022. Learning-Based Lossless Point Cloud Geometry Coding Using Sparse Tensors. In *Proceedings of the 29th IEEE International Conference on Image Processing (ICIP'22)*. Bordeaux, France.

Nguyen, D. T.; et al. 2021. Multiscale deep context modeling for lossless point cloud geometry compression. In *Proceedings of the 10th IEEE International Conference on Multimedia & Expo Workshops (ICMEW'21)*. Shenzhen, China.

Ni, J.; Zhang, K.; and Vasilakos, A. V. 2020. Security and privacy for mobile edge caching: Challenges and solutions. *IEEE Wireless Communications*, 28(3): 77–83.

Pan, Z.; Xiao, M.; Han, X.; Yu, D.; et al. 2024. patchDPCC: A Patchwise Deep Compression Framework for Dynamic Point Clouds. In *Proceedings of Conference on Artificial Intelligence, (AAAI'24)*. Vancouver, BC, Canada.

Peng, S.; Jiang, C.; Liao, Y.; Niemeyer, M.; et al. 2021. Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems*, 34: 13032–13044.

Qi, C. R.; et al. 2017. PointNet++: deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Long Beach, CA, USA.

Quach, M.; Valenzise, G.; and Dufaux, F. 2019. Learning convolutional transforms for lossy point cloud geometry compression. In *Proc. of IEEE International Conference on Image Processing (ICIP'19)*. Taipei, Taiwan.

Quach, M.; Valenzise, G.; and Dufaux, F. 2020. Improved Deep Point Cloud Geometry Compression. In *Proceedings of the IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP'20)*. Virtual Event.

Reisizadeh, A.; Farnia, F.; Pedarsani, R.; and Jadbabaie, A. 2020. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Processing Systems*, 33: 21554–21565.

Ren, Y.; Lv, Z.; et al. 2023. HCNCT: A Cross-chain Interaction Scheme for the Blockchain-based Metaverse. *ACM Transactions on Multimedia Computing, Communications and Applications*.

Rigaki, M.; and Garcia, S. 2023. A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 56(4): 1–34.

Rombach, R.; Blattmann, A.; Lorenz, D.; et al. 2022. High-resolution image synthesis with latent diffusion models. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. New Orleans, US.

Sinha, A.; Song, J.; Meng, C.; et al. 2021. D2c: Diffusion-decoding models for few-shot conditional generation. *Advances in Neural Information Processing Systems*, 34: 12533–12548.

Song, C.; and Raghunathan, A. 2020. Information leakage in embedding models. In *Proc. of ACM SIGSAC Computer and Communications Security (CCS'20)*. Virtual.

Tian, D.; Ochimizu, H.; Feng, C.; Cohen, R.; and Vetro, A. 2017. Geometric distortion metrics for point cloud compression. In *Proc. of IEEE International Conference on Image Processing (ICIP'17)*. China.

Vahdat, A.; Kreis, K.; and Kautz, J. 2021. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34: 11287–11302.

Vladimirov, I.; et al. 2022. Security and Privacy Protection Obstacles with 3D Reconstructed Models of People in Applications and the Metaverse: A Survey. In *Proc. of International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST'22)*. Ohrid, North Macedonia.

Wang, J.; Ding, D.; Li, Z.; Feng, X.; et al. 2023. Sparse Tensor-Based Multiscale Representation for Point Cloud Geometry Compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7): 9055–9071.

Wang, J.; Ding, D.; Li, Z.; and Ma, Z. 2021. Multiscale Point Cloud Geometry Compression. In *Proc. of Data Compression Conference (DCC'21)*. Virtual.

Wang, J.; Zhu, H.; et al. 2021. Lossy Point Cloud Geometry Compression via End-to-End Learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12): 4909–4923.

Xu, Y.; Liu, Z.; Tegmark, M.; et al. 2022. Poisson flow generative models. *Advances in Neural Information Processing Systems*, 35: 16782–16795.

Yi, X.; Yao, L.; and Ziyu, W. 2017. Owlii Dynamic human mesh sequence dataset.

Zeng, X.; et al. 2022. LION: Latent point diffusion models for 3D shape generation. *arXiv preprint:2210.06978*.

Zhang, J.; Chen, T.; Ding, D.; et al. 2023. G-PCC++: Enhanced Geometry-based Point Cloud Compression. In *Proc. of ACM international conference on multimedia (MM'23)*. Ottawa, Canada.

Zhou, L.; Du, Y.; et al. 2021. 3D Shape Generation and Completion through Point-Voxel Diffusion. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV'21)*. Virtual Event.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76.