Pagoda: Privacy Protection for Volumetric Video Streaming through Poisson Diffusion Model

Rui Lu*
Hong Kong Polytechnic University
Hong Kong SAR, China
csrlu@comp.polyu.edu.hk

Lai Wei*
Hong Kong Polytechnic University
Hong Kong SAR, China
cslwei@comp.polyu.edu.hk

Shuntao Zhu
Hong Kong Polytechnic University
Hong Kong SAR, China
shun-tao.zhu@connect.polyu.hk

Chuang Hu†
Wuhan University
Wuhan, Hubei, China
handc@whu.edu.cn

Dan Wang
Hong Kong Polytechnic University
Hong Kong SAR, China
csdwang@comp.polyu.edu.hk

ABSTRACT

With the increasing popularity of 3D volumetric video applications, e.g., metaverse, AR/VR, etc., there is a growing need to protect users' privacy while sharing their experiences during streaming. In this paper, we show that the existing privacy-preserving approaches for dense point clouds suffer a massive computation cost and degrade the quality of the streaming experience. We design Pagoda, a new **P**riv**A**cy-preservin**G** V**O**lumetric Vi**D**eo Stre**A**ming incorporating the MPEG V-PCC standard, which protects different domain privacy information of dense point cloud, and maintains high throughput. The core idea is to content-aware transform the privacy attribute information to the geometry domain and content-agnostic protect the geometry information by adding Poisson noise perturbations. These perturbations can be denoised through a Poisson diffusion probabilistic model on the cloud. Users only need to encrypt a small amount of high-sensitive information and achieve secure streaming. Our designs ensure the dense point clouds can be transmitted in high quality and the attackers can hardly reconstruct the original one. We evaluate Pagoda using three volumetric video datasets. The results show that Pagoda outperforms existing privacy-preserving baselines for 75.6% protection capability improvement, 4.27 times streaming quality, and 26 times latency reduction.

CCS CONCEPTS

• **Security and privacy** → **Systems security**; • **Information systems** → *Multimedia streaming*; • **Computing methodologies** → Neural networks.

*Both authors contributed equally to this research.
†Corresponding Author

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3611946

KEYWORDS

Volumetric Videos Streaming, Dense Point Clouds, Privacy-Preserving, Denoising Diffusion Model

ACM Reference Format:
Rui Lu, Lai Wei, Shuntao Zhu, Chuang Hu, and Dan Wang. 2023. Pagoda: Privacy Protection for Volumetric Video Streaming through Poisson Diffusion Model . In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3611946

1 INTRODUCTION

Social applications, like Metaverse and eSports, are encouraging users to interact and share their surroundings for immersive experiences, driving the demand for volumetric video streaming. Volumetric videos typically use the dense point cloud as the format for high precision, high resolution, and rich color representations. To efficiently transmit the dense point clouds, the ISO Moving Picture Experts Group (MPEG) proposes *video-based point cloud compression (V-PCC)* as the dense point cloud compression standards [37]. V-PCC transforms 3D point clouds into a set of 2D images consisting of Attribute Images, Geometry Images, and Occupancy Maps. It provides high compression efficiency, using well-established 2D video coding technology, and has been widely adopted by industries and companies, including Nokia-AR [36], Intel [28], and Sony [8].

Privacy concerns arise as users increasingly share volumetric videos. During transmission, the complex and vulnerable cyber environment of users makes uploaded streams easy to hijack and extract private information [13, 25]. Malicious applications can also induce users to send private videos to their servers, resulting in privacy leaks [46]. For example, Nokia-AR allows cell phones to capture volumetric videos and transmit them to the cloud for rendering via V-PCC [36]. These volumetric streams contain video copyright information and user-related sensitive information, which would be a massive loss if someone were to hijack them. However, existing state-of-the-art approaches primarily focus on addressing resource and performance challenges, neglecting privacy concerns.

Existing protection schemes for volumetric video can be categorized into hardware protection, e.g., establishing a Trusted Execution Environment (TEE) [32], data encryption, e.g., frame encryption [39] and point cloud perturbation, e.g., adding random noises and denoising [21, 55], adaptive perturbations [17, 38]. However, these techniques are unsuitable for protecting video streaming for

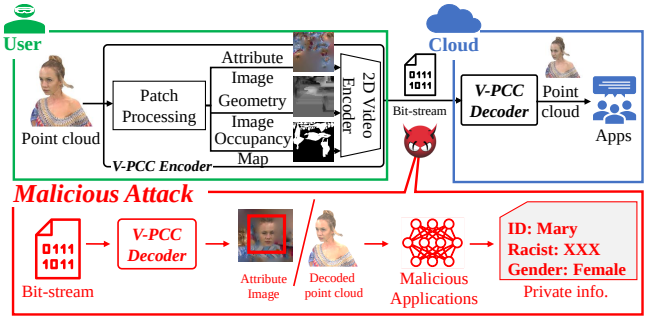


Figure 1: Workflow of V-PCC and malicious attacks.

users with limited resources. For instance, TEE demands additional hardware upgrades, while encryption approaches consume considerable computation resources. Point cloud perturbation has fewer requirements but still suffers high latency and low throughput, which is hard to apply on volumetric video streaming protection.

Meanwhile, there are numerous methods designed to protect the privacy of 2D video streaming, e.g., adaptive perturbations [48, 49], noise-denoising [10], image transformation [20], privacy de-identification [10], blurring [45], etc. However, these methods can not be applied directly to protect the privacy of point clouds because of the inherent disordered data structures and the additional geometric dimension compared to images.

Furthermore, internal functions of V-PCC codec include transforming 3D point clouds into 2D images and separating the information into attribute and geometry domains. This conversion provides us an opportunity to protect point clouds through 2D schemes. Upon investigating the performance of these 2D protection schemes on such images, we discover that they are not particularly effective. Although the target data has been transformed into 2D images, the information entropy decreases marginally. For example, neural-network-based schemes [49] still require the input of large-sized images, resulting in significant computational overhead. Consequently, the viable option is to introduce intense noise perturbation, e.g., Poisson noise [50], on the user and perform accurate denoising on the cloud.

In our design, we propose **Pagoda**, a novel **P**riv**A**cy-preservin**G** V**O**lumetric Vi**D**eo Stre**A**ming enhancement for V-PCC, that fully utilizes V-PCC compression components for privacy protection and maintains encoding efficiency. We first utilize Patch Processing in V-PCC, which segregates the attribute information, geometry information, and occupancy status in point clouds into 2D video sequences. We then protect attribute and geometry information as follows: 1) We protect the privacy attribute information by transferring them to the geometry domain. 2) We protect the geometry information by adding Poisson perturbation which will be purified on the cloud through a denoising diffusion model. 3) We protect the occupancy states and other high-sensitive information by encryption. The final outputs can still be compressed by V-PCC 2D video encoder and guarantee a low transmission overhead. We implement Pagoda on the public V-PCC repository TMC2 [24]. Finally, we evaluate Pagoda on three popular volumetric video datasets and show how it significantly outperforms existing baselines for 75.6% protection capability improvement, 4.27 times streaming quality, and 26 times latency reduction.

To the best of our knowledge, Pagoda is the first attempt to provide a privacy protection mechanism for V-PCC. Our contributions are summarized as follows:

- We analyze existing volumetric streaming protection approaches and their limitations (§ 2).
- We develop Pagoda, a novel Privacy-preserving Volumetric Video Streaming enhancement incorporating V-PCC through Privacy-preserved Patch Processing and Poisson Noise Perturbation and Purification (§ 3).
- We evaluate Pagoda on three volumetric video datasets, and it outperforms other baselines with a high privacy protection capability, streaming quality, and encoding latency (§ 4).

2 BACKGROUND AND MOTIVATION

2.1 Volumetric Video Streaming

Video-based Point Cloud Compression. The MPEG-developed standard for compressing dynamic dense point clouds, called Video-based Point Cloud Compression (V-PCC), uses 2D video codecs, e.g., AVC and HEVC, to compress 3D point clouds into 2D video sequences (see Fig. 1). As part of the compression process, Patch Processing is applied to project each point's information into Attribute Images, Geometry Images, and Occupancy Maps, which are compressed into bitstreams by a 2D video encoder and transmitted to the cloud. Upon receiving the bitstream, the cloud initiates the decompression process by decoding it back into images and then using Patch Unpacking to reconstruct the 3D point clouds.

Privacy Leakage in Volumetric Video Streaming. Malicious Attack on Streaming is a serious threat that involves unauthorized access and interception of streaming bitstreams. Attackers can modify or inject data into the stream, potentially compromising the confidentiality and integrity of the data. Attackers may include malicious servers attempting to extract users' personal information [13, 25] or criminals conducting man-in-the-middle attacks [46]. An example is shown in Fig. 1, illustrating the recovery process of a point cloud from an intercepted bitstream. The attack begins by hijacking the bitstream transmitted from the user to the cloud. The attacker then identifies the codec and employs a corresponding decoder to decompress it. Sensitive information, such as attribute information or even point clouds, can be extracted or reconstructed from the bitstream. With advanced machine learning techniques, e.g., Neural Network (NN) models, information like facial features, user demographics, and identities can be retrieved.

2.2 Motivation

We conduct a motivational experiment on protecting a volumetric video sequence for user upstreaming on Oculus Quest [22] and streaming longdress@8iv2 [15] point clouds. We study three state-of-the-art protection approaches for V-PCC: 1) VVSec [42] applies 3D adaptive perturbation predicted by NN, and 2) LION [55] adds intense Gaussian noise in point clouds, 3) SAPV [49] applies adaptive perturbation on 2D images transformed from point cloud by V-PCC encoder. We also inject intense noises for perturbation, i.e., Gaussian noise, into those 2D images.

Table 1 shows the computation time components of each protection approach. The results illustrate that the latency of NN inference and noise execution time is massive on point clouds. In contrast,

Table 1: Run time breakdown among privacy-preserving approaches evaluated by *longdress*@8iv2 [15].

Protection Scheme	Protection Applied Target	NN Inference Time (s)	Protection Execution Time (s)	V-PCC Encoding Time (s)	Total Time (s)
V-PCC only	/	/	/	1.14	1.14
VVSec [42]	point clouds	39.4	3.12	2.28	44.8
LION [55]	point clouds	/	48.66	4.59	53.25
SAPV [49]	2D images	21.01	0.48	1.55	23.04
Intense noise	2D images	/	0.89	1.18	**2.07**

the latency of NN inference on 2D images is less than that of point clouds but still requires 13.5 times encoding time. Interestingly, the noise execution time of intense random noise on 2D images is only about 0.75 times, which could be our optimal solution.

2.3 Potential Approaches.

The *Poisson Diffusion Model* is a mathematical model for image denoising. It is based on the assumption that the distribution of noise pixels in the image follows the Poisson distribution and thus can be eliminated by approximating the denoising distribution. In this paper, we investigate the potential capability of using the Poisson diffusion model for protecting privacy in point cloud streaming.

Denoising Diffusion Model (DDM) is one of the most popular generative models [26] contains two phases, the *forward* process to arbitrary noises, e.g., Gaussian noise [10], Gamma noise [6], and *reverse* process to approximate the target distribution by denoising. Although DDMs are not designed for image protection, the forward process of adding noises into clean images can be utilized as the perturbation to protect images. The perturbation can be removed accurately during the reverse process by DDMs. Typical Markov-chain noise models, e.g., Gaussian noise or Gamma noise, are not suitable for privacy protection because the noise of the forward process can be reproduced by anyone to train a general denoising model. For example, the attacker could remove the noise by 1) inputting his point clouds into the public V-PCC encoder, 2) obtaining clean 2D images as training data, and 3) adding the noise as input to train his denoising model. In contrast, *Poisson noise model* [50] is a non-Markov-chain noise model. The noisy images sampled from *Poisson noise model* with the *Poisson noise keys* can only be precisely denoised by the *Poisson denoising diffusion model* trained with the *Poisson noise keys*. Only those authorized clouds with the correct Poisson keys can train their denoising models and achieve perfect or near-perfect denoising performance.

To apply the Poisson diffusion model to protect the 2D images generated inside V-PCC encoder, we observe that 75% of the images consist of attribute information, 24% comprise geometry information, and a mere 1% are occupancy states.

Most data volume is constituted by attribute information, placing the various colors of unordered points on Attribute Images. The only privacy the attacker can hack from them is those points that project the privacy parts of point clouds, e.g., human faces and car license plates, etc., identifiable by their distinct color attributes. Moreover, the attribute information is sensitive to intense noises and jeopardizes the streaming quality [51]. If we could avoid obfuscating such attribute information, we only need to apply the Poisson noise to geometry information. Theoretically, it can reduce 2/3 of protection latency overall and improve the streaming quality.

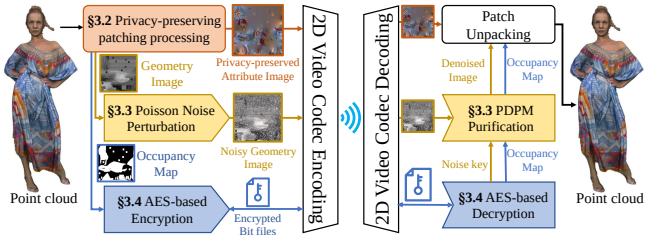


Figure 2: Workflow of protecting volumetric video streaming by Pagoda, where the black border graphics are MPEG V-PCC modules without modification, including 2D video codec Encoding/decoding and Patch Unpacking.

3 DESIGN

3.1 Design Overview

We first present the design points and a solution overview on enhancing the privacy-preserving capability of V-PCC for volumetric video streaming. There are three design points as follows:

• **Protecting the attribute information:** Attribute information stored in color Attribute Images, occupies the largest size. We introduce a Privacy-preserving Patch Processing algorithm to content-aware transform the privacy attribute information, i.e., Regions of Privacy (RoPs), into geometry information by an RoP detector. We separate and reallocate RoPs into unoccupied regions and finally obtain Privacy-preserved Attribute Images.

• **Protecting the geometry information:** Geometry Images are single-channel images to record geometry information. We introduce a *Poisson Diffusion Model Perturbation and Purification* approach to content-agnostic protect them. The diffusion forwarding process is to add random Poisson noise on the user. We develop a Poisson diffusion probabilistic model, PDPM, for the reverse process on the cloud to cancel those noises.

• **Protecting high-sensitive information:** To protect the occupancy states and others, i.e., Occupancy Maps, Poisson key index, etc, we first compress them to binary files by video codec. They are then encrypted by a secure but slow encryption approach to ensure secure transmission to the cloud.

Pagoda for Privacy-preserving Volumetric Video Streaming: In Fig. 2, the point clouds are first processed by Privacy-preserving Patch Processing (§ 3.2) and generate Privacy-preserved Attribute Images, Geometry Images, and Occupancy Maps. Geometry Images are protected by Poisson Noise Perturbation (§ 3.3), Occupancy Maps, and other high-sensitive information, i.e., Poisson noise keys, are encrypted by AES-based Encryption (§ 3.4). They are compressed by 2D video encoder and transmitted to the cloud. On the cloud, noisy Geometry Images, and encrypted high-sensitive information are recovered by PDPM Purification (§ 3.3) and AES-based Decryption (§ 3.4). They are finally reconstructed to point clouds via Patch Unpacking in V-PCC decoder.

Threat Model. We assume that the cloud has formidable computational capabilities and the codec is publicly accessible for the cloud, users, and attackers. Users obtain point clouds and compress them through V-PCC encoder with privacy-preserving mechanisms if applicable, e.g., Pagoda, before upstreaming to the cloud. The attacker can input his point clouds to the public codec in advance and

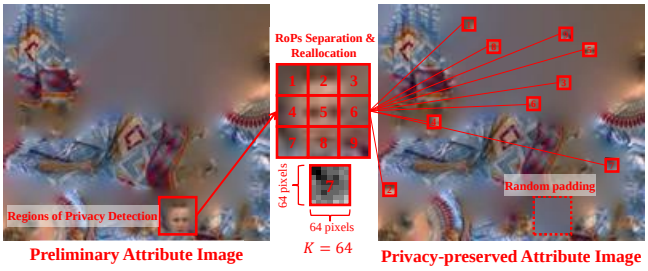


Figure 3: Workflow of Privacy-preserving Patch Processing for attribute protection. The Regions of Privacy (RoPs) are first detected by RoP Detector. Then they are separated into blocks and reallocation to the unoccupied areas.

obtain the necessary data, e.g., noised images for training denoising models. During the attack, the attacker hijacks the bitstream from victims and tries against the privacy protection methods, e.g., a denoising model to cancel noises, except violently decrypted encryption-based methods. The attack will be considered a success if the attacker 1) reconstructs the original point clouds; 2) obtains partial attribute information with privacy features, e.g., human face.

3.2 Privacy-preserving Patch Processing for Attribute Protection

The protection of attribute information involves three major steps. The first step is Preliminary Patch Generation which follows the same procedure as V-PCC Patch Processing introduced in Section 2.1, projecting the points into three 2D images, named *Preliminary Attribute Images*, *Preliminary Geometry Images*, and *Preliminary Occupancy Maps*. In the second step, we detect RoPs in the Preliminary Attribute Image through a RoP detector. In the third step, we divide them into smaller blocks and reallocate them to unoccupied areas to generate Privacy-preserved Attribute Images.

3.2.1 Regions of Privacy (RoPs) Detection. RoPs detection is designed to precisely detect and localize RoPs in the Preliminary Attribute Images through an object detection model. Our design goal is to make a balance between computational complexity and detection accuracy to achieve a practical solution for RoP detection.

• **Regions of Privacy Definition.** RoP is defined to appear in Preliminary Attribute Images and contains sensitive visual features, e.g., human faces, as shown in Fig. 3. Once they are hijacked by attackers, the privacy information is directly exposed and they no longer need to reconstruct the complete point cloud. The sensitive features include human features as presented in [27], e.g., eye, ear, face, etc., and sensitive textual, e.g., location, phone number, landmark name, etc. We label them in the volumetric datasets and create an *RoP dataset* for the training and inference of the RoP detector.

• **Detector Model and Training.** There are several object detection NN models, e.g., SSD300 [18], YOLOv5 [12], TinyYOLOv3 [33], etc. We choose TinyYOLOv3 as our detector model since it has a great performance and fast response due to its small network architecture with fewer layers and parameters. This compact design reduces computational consumption, making it suitable for deployment inside V-PCC codec. It is trained by the RoP dataset.

• **RoP Detector on Service.** The detector takes Preliminary Attribute Images as input and returns the positions of RoPs. To further reduce the computation consumption and improve the encoder throughput, we first downsample the shape of the Preliminary Attribute Image into a small scale, i.e., 256 × 256, and transfer it into grayscale through NTSC [19]. Although these operations will degrade the detection accuracy by about 12.8% and 7.5%, respectively, they will increase over 38 times the encoder throughput in total.

Algorithm 1: Privacy-preserving Patch Processing

Input: Privacy-intrusion level K;input volumetric frame: point cloud $\mathbf{P}$
Output: Geometry Image I_g, Occupancy Map I_o, Privacy-preserved Attribute Image I_a

1 $I_a, I_g, I_o \leftarrow$ *V-PCC_Patch_Processing*($\mathbf{P}$) ;
2 $I_a \leftarrow$ *downsampling*(I_a);
3 $I_a \leftarrow$ *NTSC*(I_a);
4 Regions of Privacy $\mathbf{R} \leftarrow$ ***RoP_Detector***(I_a);
5 Separate $\mathbf{R}$ into $K \times K$ size blocks $\mathbf{B} = \{(p_i; K)\} \leftarrow \mathbf{R}$;
6 Unoccupied Regions $\mathbf{U} = \{(p_i; K) | I_o(p_i; K) = \mathbf{0}\}$;
7 **for** $(p_i; K) \in \mathbf{B}$ **do**
8 Random select $(p_j; K) \in_R \mathbf{U}$, $\mathbf{U} \leftarrow \mathbf{U} - \{(p_j; K)\}$;
9 $I_a(p_j; K) \leftarrow I_a(p_i; K)$, $I_a(p_i; K) \leftarrow$ *random* ;
10 $I_g(p_j; K) \leftarrow I_g(p_i; K)$, $I_g(p_i; K) \leftarrow$ *random* ;
11 $I_o(p_j; K) \leftarrow \mathbf{1}$, $I_o(p_i; K) \leftarrow \mathbf{0}$;
12 Output I_a^t, I_g^t, I_o^t.

3.2.2 Privacy-preserved Attribute Image Generation.

• **RoPs Separation.** After locating RoPs from the RoP detector, we separate them into $K \times K$ blocks, denoted by $\mathbf{B}$, where K is the privacy-intrusion level designated by the user, p_i is the central coordinate of block i, as introduced in Algorithm 1 line 5. Basically, an RoP block is defined as *tiny* if its length is less than 64 [14]. These tiny blocks are beyond the recognition capabilities of human eyes and NN models [29]. Lower values of K correspond to greater levels of privacy protection, as RoPs are separated into smaller blocks. Note that the value of K should be larger than the custom precision parameter B_0 of Occupancy Maps [37]. The lowest value of K implies that each pixel constitutes a block. Moreover, reducing K will take more blocks resulting in an increasing number of patches.

• **RoPs Reallocation.** We first fast find Unoccupied Regions $\mathbf{U}$ in the Preliminary Attribute Image. Note that $I(p; k)$ represents a square block of center coordinate p with length k on image I. One of the efficient ways is to traverse the Occupancy Map to locate those $\mathbf{0}$-value regions with the size of $K \times K$, as described in Algorithm 1 line 6, and get their correlated coordinates in Preliminary Attribute Images. The blocks are then randomly reallocated to these Unoccupied Regions and synchronized these changes in Preliminary Geometry Images and Preliminary Occupancy Maps, as described in Algorithm 1 line 8-12. We also need to blur RoP blocks by assigning random values to hide their information. Once all RoPs are separated and moved, we obtain Privacy-preserved Attribute Images, Geometry Images, and Occupancy Maps.

For example, a human face is detected in a Preliminary Attribute Image, as shown in Fig 3. We roughly set privacy-intrusion level $K = 64$ so that the human face is divided into 9 blocks, whose length is 64 pixels. We then randomly reallocate them into unoccupied regions and generate a Privacy-preserved Attribute Image. If the attacker tries to recover the human face in the Privacy-preserved Attribute Images, he has to look up the corresponding Geometry Images to reconstruct the relationship among disordered blocks. In other words, the privacy attribute information in the Attribute Image is *transformed* into the geometry domain.

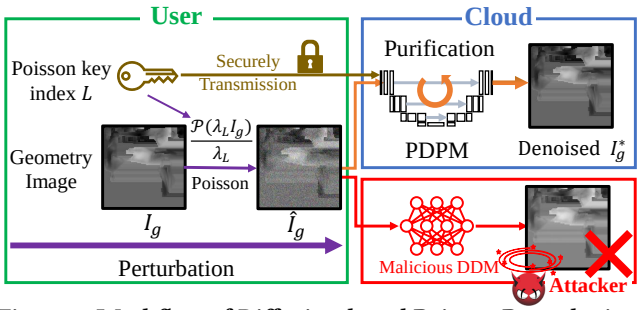


Figure 4: Workflow of Diffusion-based Poisson Perturbation and Purification for geometry protection.

3.2.3 Patch Unpacking on the Cloud. The unpacking patch process on the cloud is identical to that of V-PCC decoder without Privacy-preserving Patch Processing. It is because we only change the pixel positions of RoPs without modifying their values. Therefore, an unmodified V-PCC decoder can successfully unpack the Privacy-preserved Attribute Images back to the original point cloud with corresponding Geometry Images and Occupancy Maps.

3.3 Poisson Diffusion Model Perturbation and Purification for Geometry Protection

After the privacy attribute information in Attribute Images is transformed into the geometry domain reordered in Geometry Images as described in Section 3.2, we introduce a Poisson diffusion model to content-agnostic protect the geometry information including two processes as shown in Fig. 4: 1) the *forward process* is used as the *perturbation* to add random and intense Poisson noise into the Geometry Images on the user side; 2) the *reverse process* is used as the *purification* to cancel those intense noises on the cloud side.

3.3.1 Forward Process for Perturbation. In the forward process, the Geometry Images are injected with random and intense Poisson noise for content-agnostic protection as shown in Fig. 4. It aims to add the perturbation, which ensures when the attacker intercepts these noisy images, they are failed to denoise them, however, the cloud server can near perfectly cancel them. A Poisson noise model [50] is first defined to sample noisy images as follows:

$$\mathbf{x}_t|\mathbf{x}_0 \sim \frac{\mathcal{P}(\lambda_t \mathbf{x}_0)}{\lambda_t}, t = 1, 2, \dots \tag{1}$$

where $\mathcal{P}(\lambda_t \mathbf{x}_0)$ is a Poisson distribution with parameters $\lambda_t \mathbf{x}_0$ and $\mathbf{x}_0$ is the input clean image. *Note that t here is NOT the time stamp of the volumetric video sequences, but the denoising step running on the cloud.* The greater of t will not affect the perturbation computation cost on users. Moreover, the latency of sampling a noisy single-channel image from a Poisson distribution is ignorable.

Before streaming starts, Poisson keys Λ are generated randomly and accessed by users and the cloud only. Λ is a sequence with N items, defined as $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$ and satisfies the condition of $0 < \lambda = \lambda_N < \lambda_{N-1} < \cdots < \lambda_1 < \lambda_0 \to \infty$, where λ is a customized constant. Since Eq. 1 is true if $\lambda_t - \lambda_{t+1} > 0$ [50], making Λ has to be a monotonically decreasing sequence. Although the Poisson noise model is not Markov-chain, its forward process can be reformulated to complete a full diffusion process [50] as:

$$q(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_{t+1:N}) = q(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_{t+1}) \tag{2}$$

The user should first randomly select a Poisson key index L from $\{1, 2, \dots, N\}$. The noisy Geometry Image is obtained by sampling

$$\hat{I}_g = \frac{1}{\lambda_L}\mathcal{P}(\lambda_L I_g). \tag{3}$$

The Poisson key index L is required to send to the cloud with $\hat{I}_g$, which will describe in Section 3.4.

3.3.2 Poisson Diffusion Probabilistic Model Design and Training. Similar to Equation 2, the reverse process of Poisson denoising can be approximated by follows:

$$p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_N) \sim \frac{1}{\lambda_t}(\lambda_{t+1}\mathbf{x}_{t+1} + \mathcal{P}((\lambda_t - \lambda_{t+1}) f_\Lambda(\mathbf{x}_{t+1}, \mathbf{x}_N, t+1; \theta))) \tag{4}$$

where $f_\Lambda(\mathbf{x}_{t+1}, \mathbf{x}_N, t; \theta)$ is a Poisson diffusion probabilistic model (PDPM), with parameters θ with $\mathbf{x}_{t+1}, \mathbf{x}_N$ inputs and train with Poisson key Λ. We applied a UNet [34], as shown in Fig. 4, to support mapping noised input to reverse process parameters where UNet is tailored for 2D video frames. f_Λ can be trained as follows:

$$\theta^* = \underset{f_\Lambda, \theta}{argmin}\, \mathbb{E}_q \|f_\Lambda(\mathbf{x}_{t+1}, \mathbf{x}_N, t+1; \theta) - \mathbf{x}_0\|_2^2. \tag{5}$$

And the loss function is approximated as follows:

$$L = \mathbb{E}_q \sum_{t=0}^{t=N-1} \|f_\Lambda(\mathbf{x}_{t+1}, \mathbf{x}_N, t+1; \theta) - \mathbf{x}_0\|_2^2. \tag{6}$$

The training process is introduced in detail in Algorithm 2. The training dataset of geometry images can be easily generated from volumetric datasets through V-PCC encoder.

Algorithm 2: Poisson Noise Purification Models Training

Input: Geometry Images dataset $\{\mathbf{x}_0\}$, Denoising Model $f_\Lambda(\cdot, \cdot, \theta)$ with parameters θ, Poisson keys $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$.
Output: Well-trained parameters θ^*.

```
while θ is not converged do
    Random select x_0 from {x_0} uniformly;
    Random select t from {1, 2, ..., N − 1} uniformly;
    Sample x_N from P(λ_N x_0)/λ_N;
    Sample x_t from (1/λ_t)(λ_{t+1} x_{t+1} + P((λ_t − λ_{t+1}) x_0));
    Compute gradient ∇_θ ||f_Λ(x_t, x_N, t; θ) − x_0||_2^2;
    Update θ* ← θ;
Output θ*;
```

3.3.3 Reverse Process for Purification. On the cloud, it receives the Poisson key index L and noisy Geometry Images $\hat{I}_g$ as the $\mathbf{x}_L$ as shown in Fig. 4. We then start denoising via PDPM f_Λ from $\mathbf{x}_L$ to x_1. For each denoising step t, We sample τ_t from $\mathcal{P}((\lambda_t - \lambda_{t+1}) f_\Lambda(\mathbf{x}_{t+1}, \mathbf{x}_N, t+1; \theta^*))$, and $\mathbf{x}_t = \frac{1}{\lambda_t}(\lambda_{t+1}\mathbf{x}_{t+1} + \tau_t)$, where $t = L-1, \dots, 1$. The denoised Geometry Image I_g^* will be obtained at the last round of PDPM inference as:

$$I_g^* = f_\Lambda(\mathbf{x}_1, \mathbf{x}_L, 1; \theta^*) \tag{7}$$

We further refine the I_g^* according to occupancy states. We utilize the intersection between the Geometry Image and its corresponding Occupancy Map to determine whether the pixels on the Geometry Image are noise. The refined Geometry Image is defined as:

$$I_g^{**} = \sum^{p_i \in \mathbf{B}} I_g^*(p_i; K) j_i, \quad j_i = \begin{cases} \mathbf{0}, & if\, I_o(p_i; K) = \mathbf{0} \\ \mathbf{1}, & if\, I_o(p_i; K) = \mathbf{1} \end{cases} \tag{8}$$

3.4 Encryption for High-sensitive Information Protection

In our proposed design, the critical point of geometry information protection is the Poisson keys index L. Unless the attackers can access each Poisson key for each Geometry Image, they cannot train their own denoising model and reverse the geometry information perfectly. Another kind of information we need to protect securely is occupancy status in Occupancy Maps. They can potentially help the attacker against the Poisson noise because they can distinguish noisy and clean geometry pixels as described in Eq. 8. Overall, we securely transmit Occupancy Map I_o, Poisson keys index L, and other compression metadata to the cloud utilizing a V-PCC encoder and AES-128 encryption. There are only about 1% of data required encryption among all transmission data. We apply an AES-128 modified encryption [35] to guarantee the security of transmission. It has about 500 Bytes/ms throughput for bin file encryption, which is far below the bottleneck threshold.

Encryption on Users: Before streaming, the user is required to generate a 128-bit secret key, which should be securely transmitted to the cloud by using public key protocols before the start of the entire transmission. During transmission, the binary file undergoes padding to make it divisible into 128-bit blocks. Subsequently, the AES generates an Initialization Vector (IV). The AES cipher employs the secret key and IV to encrypt each block. Each operation output serves as the IV for the subsequent block. The blocks and the IV are stored in a new binary file and transmitted to the cloud.

Decryption on the Cloud: The cloud receives the secret key before starting transmission. After receiving the encrypted binary files, decryption is performed utilizing the secret key and the most recent IV. The decrypted file is subsequently processed by V-PCC decoder, adhering to standard procedures for recovering the binary file into Occupancy Maps, Poisson keys index, and others.

4 IMPLEMENTATION AND EVALUATION

In this section, we present the implementation of Pagoda and evaluate its performance aiming to answer the following questions:

- How does our design incorporate inside V-PCC standard and communicate with other components? (§ 4.1)
- How does our design compare to the existing privacy-preserving mechanisms applied to V-PCC? (§ 4.3)
- How do the noise-denoise models affect and contribute to the overall performance? (§ 4.4)

4.1 Implementation

We establish Pagoda on V-PCC open-source repository TMC2 [24]. We revised the codes to support the FFMPEG 2D video codec and leverage hardware acceleration. We train our RoPs Detector and PDPM using Pytorch [31]. We integrate LibTorch [3] to import the well-trained NN models to be employed by TMC2 in C++. We program our lightweight encryption based on Crypto++ [1].

Training Details: *RoPs Detector* use Adam optimizer with an initial learning rate of 10e-4 with a 512 batch size for training. *PDPM* uses Adam with an initial learning rate of 2e-5 for training. The batch size is 64 where $N = 40$ and $\lambda = 0.1$.

4.2 Experimental Setup

4.2.1 Testbed. We evaluated Pagoda on a user-cloud environment to simulate a real volumetric video streaming scenario. On the user, we use an Oculus Quest [22] 6GB memory with H.264/H.265 video encoding hardware acceleration. On the cloud, we used a workstation featuring dual powerful NVidia RTX 4090 GPUs, and an Intel i9 CPU, providing exceptional video decoding and NN computation capabilities. They are connected by a wired cable.

4.2.2 Volumetric datasets. We use three volumetric video datasets to evaluate: 1) *8i Voxelized Full Bodies (8iv2)* [15] contains four dynamic point cloud sequences over 5.5GB, including longdress, loot, red-and-black, and soldier; 2) *Owlii Dynamic Human Textured (Owlii)* [53] includes basketball-player, dancer, exercise, and model four sequences over 39.5 GB; 3) *Microsoft Voxelized Upper Bodies (MVUB)* [5] contains ten human upper bodies sequences over 10GB.

4.2.3 Baselines. We compare Pagoda with four state-of-the-art privacy-preserving methods. Most are either open-source or easily modifiable, making them suitable as benchmarks.

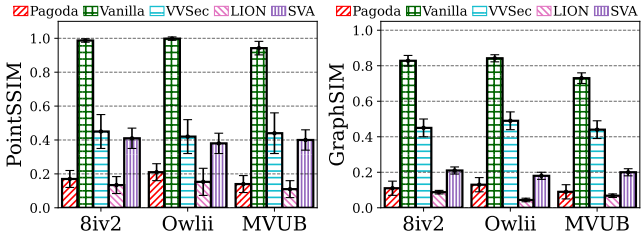
- ***Vanilla*** is a *non-privacy-preserving* approach to encode, transmit, and decode point clouds via V-PCC. It is the lower bound for privacy protection and the upper bound for streaming quality.
- ***VVSec*** [42] creates *3D adaptive adversarial perturbations* by an NN model to obfuscate the point cloud, without denoising process. We revised its loss function to optimize privacy leakage and streaming distortion instead of against face authentication attacks.
- ***LION*** [55] applies a typical 3D-DDM for point cloud generation. Instead of transferring point clouds to another domain, it is revised to add *Gaussian Noise*, denoise, and regenerate the original point cloud. We run adding noise on the user, transmitting the perturbed point cloud through V-PCC, and denoising on the cloud.
- ***SVA*** [49] applies *2D sparse perturbation* to protect 2D video sequences through reinforcement learning without denoising process. We add SVA noises to Attribute Images, Geometry Images, and Occupancy Maps from V-PCC Patch Processing.

4.2.4 Evaluation Metrics. We evaluate the performance of Pagoda and baselines from four aspects: privacy leakage, streaming quality, rate distortion, and latency. We list the metrics below:

- ***Privacy Leakage Metric:*** Since we do not predefine the identity or objectives of attackers to obtain what kinds of privacy in the threat model, the quantization of a successful attack is counted by *comparing the general similarity* between the point cloud reconstructed from the interception bitstream to the original one by PointSSIM [2] and GraphSIM [52] in both attribute and geometry.
- ***Streaming Quality & Rate-Distortion Metric:*** We employ two SOTA metrics, Point Cloud Quality Metric (PCQM) [23] and point-to-plane Peak Signal-to-Noise Ratio (p2plane-PSNR) [43] that provide an insightful representation of the rate-distortion trade-off that effectively demonstrate the efficacy of our designs.
- ***Latency Metric:*** We adopt the encoding time per frame of Pagoda and baselines to evaluate the efficiency of the user upstreaming. Lower values implicit a better throughput for an encoder.

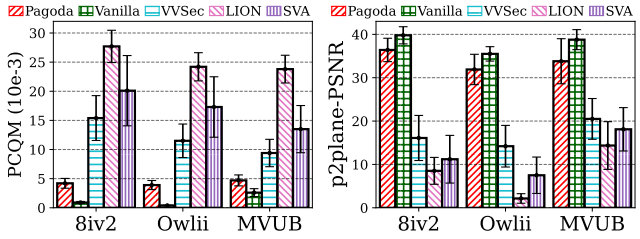
4.3 Overall Performance

4.3.1 Improvement of Privacy Protection. We evaluate the privacy protection performance of Pagoda, compared to baselines on three



(a) PointSSIM (↓) (b) GraphSIM (↓)

Figure 5: Average privacy leakage in different baselines and datasets. Larger values of both metrics indicate a higher similarity, i.e., higher privacy leakage.



(a) PCQM (↓) (b) p2plane-PSNR (↑)

Figure 6: Average streaming quality under lossless rate in different baselines and datasets. Smaller PCQM values or larger p2plane-PSNR indicate a higher streaming quality.

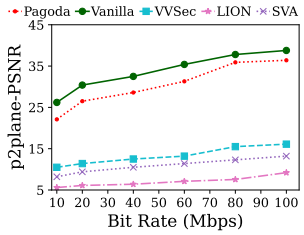


Figure 7: Rate-distortion in *longdress*@8iv2 sequence.

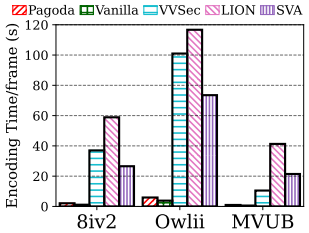


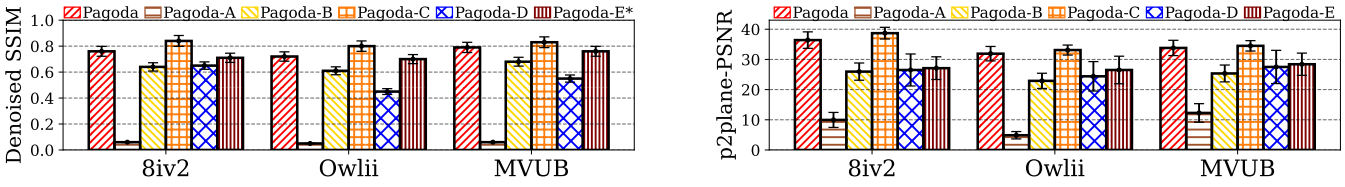
Figure 8: Average encoding time (↓) under lossless rate.

datasets. Fig. 5 shows the average PointSSIM and GraphSIM of point cloud sequences reconstructed by the attacker. Both metrics depict the level of privacy leakage. Here, Vanilla gets the highest privacy leakage because, without protection, the attacker can perfectly reconstruct the bitstream to the original point clouds. Pagoda reduces 82.8% PointSSIM and 86.7% GraphSIM than Vanilla in 8iv2. Moreover, Pagoda reduced 62.2% PointSSIM, 75.6% GraphSIM compared to VVSec, and 58.5%, 47.6% to SVA in 8iv2. Since both VVSec and SVA do not have a purification process on the cloud, they are constrained not to add too much noise to disturb the viewer on the cloud side. Note that Pagoda does not outperform LION, as LION has about 26.9% and 25.1% less privacy leakage in terms of PointSSIM and GraphSIM, respectively. This is because LION adds intense noise directly on point clouds, however, this approach comes at the cost of high computation latency, which is unsuitable for streaming. These results are consistent in the other datasets.

4.3.2 Improvement of Streaming Quality. We evaluate the streaming quality by computing two full-reference 3D video quality assessment metrics. V-PCC is set to lossless mode here. See Fig. 6. We take the p2plane-PSNR score on the 8iv2 dataset as an example. Here Vanilla serves as the upper bound because it has no privacy protection. Pagoda demonstrates only an 8.5% decline in p2plane-PSNR compared to Vanilla, which is the minimal quality degradation in all privacy protection schemes. Although intense noises are added to geometry information, an accurate and efficient PDPM is designed to cancel them. Pagoda has a better performance than VVSec, about 55.8%. It is because VVSec does not have a denoising module after receiving. Pagoda outperforms SVA and LION with a more considerable marginal improvement, achieving 2.26 times, 4.27 times higher p2plane-PSNR. Compared to Pagoda, SVA adds perturbations to both geometry and attribute images in contrast to the geometry-only approach in Pagoda. The noise in different channels is amplified during the decoding and 3D reconstruction. As for LION, it adds Gaussian noises on point clouds, but existing 3D denoising models are hard to cancel Gaussian noise perfectly. Similar results are observed in the other datasets and metrics. In conclusion, Pagoda enhances the streaming quality by adding perturbations to geometry information without modifying attribute information and accurately purifying them through PDPM.

4.3.3 Improvement of Rate Distortion. To thoroughly investigate the rate-distortion characteristics among Pagoda and baselines, we choose one trace, *longdress* from the 8iv2 dataset to observe rate distortion curve, as shown in Fig. 7. Initially, we observe that Vanilla exhibits the highest PCQM and p2plane-PSNR across all bit rates. We take p2plane-PSNR as an example. It achieves a PSNR of 26.5dB at 10Mbps and increases with a steep slope to a maximum of 36.4dB at 100Mbps. Pagoda shows a similar trend to Vanilla in terms of slope, but due to the added 2D noise on Geometry Images, it still degrades 15.6% at the Bit rate of 10Mbps, and the gap shrinks to 6.14% at the 100Mbps. However, the other three baselines demonstrate a notable loss on PNSR, with an average gap of 15. These noises handicapped V-PCC encoder's temporal prediction compression, leading to poor Rate-Distortion characteristics.

4.3.4 Improvement of Latency Reduction. We evaluate the compression latency of Pagoda and baselines using encoding time per frame under a lossless transmission rate. We take 8iv2 as an example, Vanilla requires 1.14 s to encode one frame. Pagoda demonstrates only 2.1 s in encoding time compared to Vanilla, which is the minimal latency increase in all schemes. LION requires over 58 s to protect a frame which is 26 times more than Pagoda due to the considerable computational resources, e.g., I/O, memory to apply to add perturbation on point clouds. VVSec and SVA take 37 s and 26.6 s, which are 17 and 12 times over Pagoda because it requires predicting adaptive perturbations before encoding through complex NN models. Similar results can be found in the other datasets. In summary, Pagoda has favorable latency performance and high throughput because of its implementation of a lightweight, high-throughput detection model for attribute information protection and the intense noise perturbation into geometry information, which is a low-resource-consuming operation on 2D images. Another reason is that the most computational operation is to encrypt the high-sensitive information, but the data proportion requiring encryption is small, further contributing to Pagoda's overall efficiency.



(a) SSIM (↑) (b) p2plane-PSNR (↑)

Figure 9: Ablation study of perturbed noise and purification effects. Pagoda-E* measures SSIM between the denoised and original Attribute Images while others measure that of Geometry Images.

4.4 Ablation Study

We also explore the efficacy of each component of Pagoda for a better understanding of their contributions. We utilize five breakdown versions, *Pagoda-A* to *Pagoda-E*. Pagoda-A removes the denoising mechanism and Pagoda-B randomly assigns Poisson noise keys to simulate attackers' guessing keys. Pagoda-C replaces the Poisson noise with Gaussian noise on the user and denoises them using DDIM [40] on the cloud. Pagoda-D employs a non-machine-learning denoising method [30] and Pagoda-E replaces the RoPs detection by applying Poisson noise perturbation on both Attribute and Geometry Images. We investigate the denoising effect, evaluated by SSIM [47] between noisy and denoised images in Fig. 9(a), where higher SSIM represents better purification performance.

We also use p2plane-PSNR to compare the decoded point clouds reconstructed from denoised images to the original point clouds as shown in Fig. 9(b). We take the 8iv2 dataset as an example. Pagoda-A underperforms Pagoda in SSIM and p2plane-PSNR only 8.4% and 27.1%, indicating the necessity to cancel intense noises in geometry information. Pagoda surpasses Pagoda-B in SSIM and p2plane-PSNR by 19.5% and 40.4%, demonstrating the challenge of recovering geometry without Poisson Noise Keys and validating that our protection mechanism is effective. Pagoda has similar SSIM and p2plane-PSNR values to Pagoda-C, about 11.5% and 6.3% reduction. It shows that Gaussian noise is easily recovered using DDIM by anyone. The attackers can exploit by training their DDIMs, reducing geometric protection capability. Pagoda outperforms Pagoda-D by approximately 14.2% in SSIM and 37.6% in p2plane-PSNR, as traditional denoising approaches cannot perfectly cancel noise, affecting streaming quality. Lastly, Pagoda outperforms Pagoda-E by approximately 5.2% in attribution SSIM but 34.3% in p2plane-PSNR.

5 RELATED WORK

In the research literature, Pagoda falls into a *privacy-preserving point cloud streaming* through attribute information transformation and a Poisson noise perturbation for geometry information protection.

Perturbation-based Privacy Protection. Recently, there has been the exploration of perturbation-based video privacy protection schemes [4, 20, 44]. These schemes change the content of the video before transmitting it over the Internet. For example, Vepakomma et al. [44] proposed an algorithm to inject noise into the CNN intermediate representations of a video to avoid a reconstruction attack. SAPV [49] proposes a sparse adversarial perturbation-based privacy protection scheme, which only perturbs a small portion of frames to fool the attacker's CNN model effectively. In comparison, Lu et al. [20] proposed transforming the video frames using a policy-based design There are also object inpainting [4][16][56] and object replacement[41][7] approaches, which protect privacy by removing or replacing the sensitive object in a video frame. However, studies along this line have rarely touched the point of 3D volumetric videos and cannot directly migrate to this new format.

Privacy-preserving Point Cloud Streaming. VVSec [42] proposed a privacy-preserving volumetric video streaming method through the benign use of adversarial perturbation. They design a content-aware adversarial perturbation generator to maximize the protection level while minimizing quality loss. Our work differs from it as we consider a wide spectrum of attacks instead of a single threat. LION [55] propose a general-purpose diffusion model for the static 3D point cloud, which can also be aligned to protect dense point cloud sequences. However, the LION injects Gaussian noise, when using it on a video, it takes a considerable amount of computing time, and canceling the noise is an even more intensive process. There are also 3D object inpainting-based methods[11, 54] and replacement methods[9]. To the best of our knowledge, our work is the first privacy-preserving point cloud streaming system tailored for the dense dynamic point cloud.

6 CONCLUSION

In conclusion, this paper presents Pagoda, a novel privacy-preserving mechanism for volumetric video streaming incorporating the MPEG V-PCC standard. Pagoda content-aware transforms privacy attribute information to the geometry domain before protecting geometry information using Poisson noise perturbations. The perturbations can be denoised on the cloud through a new Poisson diffusion probabilistic model, where the Poisson Distribution parameter plays the role of denoising key. Our design ensures that dense point clouds can be transmitted high quality to authorized cloud servers while blocking attackers from reconstructing the original information. Pagoda outperforms other approaches regarding privacy protection capability, encoding throughput, and streaming quality. Overall, our work has significant implications for applications, e.g., telemedicine and remote education, where sensitive information needs to be protected while maintaining high-quality streaming.

ACKNOWLEDGEMENT

This work is supported by RGC GRF 15209220, 15200321, 15201322, ITF ITS/056/22MX, CRF C5018-20G, and support from ITC via project "Smart Railway Technology and Applications" (No. K-BBY1). Chuang Hu 's work is supported by the Fundamental Research Funds for the Central Universities (2042023kf0132).

REFERENCES

[1] Crypto++ Library 8.7. [n. d.]. Free C++ Class Library of Cryptographic Schemes. https://www.cryptopp.com/

[2] Evangelos Alexiou and Touradj Ebrahimi. 2020. Towards a point cloud structural similarity metric. In *Proc. of IEEE International Conference on Multimedia and Expo Workshops (ICMEW'20)*. LA, CA, USA.

[3] PyTorch C++ API. [n. d.]. PyTorch C++ API documentation. https://pytorch.org/cppdocs/

[4] Frank Cangialosi, Neil Agarwal, Venkat Arun, et al. 2022. Privid: Practical, Privacy-Preserving Video Analytics Queries. In *Proc. of USENIX Symposium on Networked Systems Design and Implementation (NSDI'22)*. Renton, WA, USA.

[5] Loop Charles, Cai Qin, et al. [n. d.]. Microsoft Voxelized Upper Bodies - A Voxelized Point Cloud Dataset. https://plenodb.jpeg.org/pc/microsoft

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

[7] Sandeep Dsouza, Victor Bahl, Lixiang Ao, et al. 2020. Amadeus: Scalable, Privacy-Preserving Live Video Analytics. *arXiv preprint:2011.05163* (2020).

[8] Danillo Graziosi, Alexandre Zaghetto, Ali Tabatabai, and Vladyslav Zakharchenko. 2021. Synchronization of decoded frames before point cloud reconstruction. US Patent App. 17/066,434.

[9] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, et al. 2021. Pct: Point cloud transformer. *Computational Visual Media* 7 (2021), 187–199.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

[11] Wei Hu, Zeqing Fu, and Zongming Guo. 2019. Local frequency interpretation and non-local self-similarity on graph for point cloud inpainting. *IEEE Transactions on Image Processing* 28, 8 (2019), 4087–4100.

[12] Glenn Jocher. 2020. *YOLOv5 by Ultralytics*. https://doi.org/10.5281/zenodo.3908559

[13] Navid Ali Khan, Sarfraz Nawaz Brohi, and NZ Jhanjhi. 2020. UAV's applications, architecture, security issues and attack scenarios: A survey. In *Proc. of Intelligent Computing and Innovation on Data Science (ICTIDS'19)*. Petaling Jaya, Malaysia.

[14] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, et al. 2019. Augmentation for small object detection. *arXiv preprint:1902.07296* (2019).

[15] Maja Krivokuca, Philip A Chou, and Patrick Savill. 2018. 8i voxelized surface light field (8iVSLF) dataset. (July 2018).

[16] Qinya Li, Zhenzhe Zheng, Fan Wu, and Guihai Chen. 2020. Generative adversarial networks-based privacy-preserving 3D reconstruction. In *Proc. of IEEE/ACM International Symposium on Quality of Service (IWQoS'20)*. Virtual Event.

[17] Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. 2021. Pointguard: Provably robust 3d point cloud classification. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. Virtual Event.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, et al. 2016. Ssd: Single shot multibox detector. In *Proc. of European Conference on Computer Vision (ECCV'16)*. Amsterdam, Netherlands.

[19] Juwei Lu et al. 2009. On conversion from color to gray-scale images for face detection. In *Proc. of IEEE/CVF IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'09)*. Miami, FL.

[20] Rui Lu, Siping Shi, Dan Wang, Chuang Hu, and Bihai Zhang. 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.

[21] Shitong Luo and Wei Hu. 2021. Score-based point cloud denoising. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV'21)*. Virtual Event.

[22] Meta. [n. d.]. Oculus Quest II. https://www.meta.com/quest/quest-pro/

[23] Gabriel Meynet, Yana Nehmé, Julie Digne, 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.

[24] MPEGGroup. [n. d.]. GitHub - MPEGGroup/mpeg-pcc-tmc2: Video codec based point cloud compression. https://github.com/MPEGGroup/mpeg-pcc-tmc2

[25] Jianbing Ni, Kuan Zhang, and Athanasios V Vasilakos. 2020. Security and privacy for mobile edge caching: Challenges and solutions. *IEEE Wireless Communications* 28, 3 (2020), 77–83.

[26] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *Proc. of International Conference on Machine Learning (ICML'21)*. Virtual.

[27] Tribhuvanesh Orekondy, Mario Fritz, et al. 2018. Connecting pixels to privacy and utility: Automatic redaction of private information in images. In *Proc. of IEEE/CFV International Conference on Computer Vision (CVPR'18)*. SLC,UT,USA.

[28] Ozgur Oyman. 2021. Methods for timed metadata priority rank signaling for point clouds. US Patent App. 17/032,630.

[29] F Ozge Unel, Burak O Ozkalayci, and Cevahir Cigla. 2019. The power of tiling for small object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'19)*. Long Beach, CA.

[30] Jiahao Pang and Gene Cheung. 2017. Graph Laplacian regularization for image denoising: Analysis in the continuous domain. *IEEE Transactions on Image Processing* 26, 4 (2017), 1770–1785.

[31] Adam Paszke, Sam Gross, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[32] Rishabh Poddar, Ganesh Ananthanarayanan, Srinath Setty, Stavros Volos, et al. 2020. Visor: Privacy-preserving video analytics as a cloud service. In *Proc. of USENIX Conference on Security Symposium (2020)*. Virtual Event.

[33] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint:1804.02767* (2018).

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI'15)*. Munich, Germany.

[35] Rasool S Salman, Alaa K Farhan, and Ali Shakir. 2022. Lightweight modifications in the Advanced Encryption Standard (AES) for IoT applications: a comparative survey. In *Proc. of IEEE International Conference on Computer Science and Software Engineering (CSASE'22)*. Duhok, Kurdistan Region, Iraq.

[36] Sebastian Schwarz and Mika Pesonen. 2019. Real-time decoding and AR playback of the emerging MPEG video-based point cloud compression standard. *Nokia Technologies; IBC: Helsinki, Finland* (2019).

[37] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Zhu Li, et al. 2018. Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2018), 133–148.

[38] Zhenbo Shi, Zhi Chen, Zhenbo Xu, Wei Yang, Zhidong Yu, and Liusheng Huang. 2022. Shape Prior Guided Attack: Sparser Perturbations on 3D Point Clouds. In *Proc. of Conference on Artificial Intelligence (AAAI'22)*. Montreal, Canada.

[39] Warit Sirichotedumrong, Takahiro Maekawa, Yuma Kinoshita, and Hitoshi Kiya. 2019. Privacy-preserving deep neural networks with pixel-based image encryption considering data augmentation in the encrypted domain. In *Proc. of IEEE International Conference on Image Processing (ICIP'19)*. Taipei, Taiwan.

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

[41] Qianru Sun, Ayush Tewari, Weipeng Xu, Mario Fritz, et al. 2018. A hybrid model for identity obfuscation by face replacement. In *Proc. of the European conference on computer vision (ECCV'18)*. Munich, Germany.

[42] Zhongze Tang, Xianglong Feng, Yi Xie, Huy Phan, et al. 2020. VVSec: Securing Volumetric Video Streaming via Benign Use of Adversarial Perturbation. In *Proc. of ACM International Conference on Multimedia(MM'20)*. SEA, WA, USA.

[43] Dong Tian et al. 2017. Geometric distortion metrics for point cloud compression. In *Proc. of IEEE International Conference on Image Processing (ICIP'17)*. China.

[44] Praneeth Vepakomma, Abhishek Singh, et al. 2020. NoPeek: Information leakage reduction to share activations in distributed deep learning. In *Proc. of IEEE International Conference on Data Mining Workshops (ICDMW'20)*. Sorrento, Italy.

[45] Nishant Vishwamitra et al. 2017. Blur vs. block: Investigating the effectiveness of privacy-enhancing obfuscation for images. In *Proc. of IEEE/CVF International Conference on Computer Vision Workshops (CVPRW'17)*. Honolulu, HI, USA.

[46] Ivaylo Vladimirov 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)*. Niš, Serbia.

[47] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *Proc. of IEEE Asilomar Conference on Signals, Systems & Computers (ACSSC'03)*. Pacific Grove, CA, USA.

[48] Xingxing Wei, Huanqian Yan, and Bo Li. 2022. Sparse black-box video attack with reinforcement learning. *International Journal of Computer Vision* 130, 6 (2022), 1459–1473.

[49] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. 2019. Sparse adversarial perturbations for videos. In *Proc. of AAAI'19*. Honolulu, HI, USA.

[50] Yutong Xie, Minne Yuan, Bin Dong, and Quanzheng Li. 2023. Diffusion Model for Generative Image Denoising. *arXiv preprint:2302.02398* (2023).

[51] Jinrui Xing, Hui Yuan, Chen Chen, and Tian Guo. 2022. Wiener Filter-Based Point Cloud Adaptive Denoising for Video-based Point Cloud Compression. In *Proc. of International Workshop on Advances in Point Cloud Compression, Processing and Analysis (APCCPA'22)*. Lisbon,Portugal.

[52] Qi Yang, Zhan Ma, Yiling Xu, Zhu Li, and Jun Sun. 2020. Inferring point cloud quality via graph similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 6 (2020), 3015–3029.

[53] Xu Yi, Lu Yao, and Wen Ziyu. 2017. Owlii Dynamic human mesh sequence dataset.

[54] Yikuan Yu et al. 2020. Point Encoder GAN: A deep learning model for 3D point cloud inpainting. *Neurocomputing* 384 (2020), 192–199.

[55] Xiaohui Zeng et al. 2022. LION: Latent Point Diffusion Models for 3D Shape Generation. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS'22)*. New Orleans, Louisiana, USA.

[56] Bihai Zhang, Siping Shi, Dan Wang, and Chuang Hu. 2022. EPC: a video analytics system with efficient edge-side privacy control. In *Proc. of Workshop on Mobility in the Evolving Internet Architecture (MobiArch'22)*. Sydney, Australia.