User configurable 3D object regeneration for spatial privacy

Arpit Nama∗
University of Sydney
Sydney, Australia
anam9745@uni.sydney.edu.au

Amaya Dharmasiri∗
University of Moratuwa
Colombo, Sri Lanka
170131R@uom.lk

Kanchana Thilakarathna
University of Sydney
Sydney, Australia
kanchana.thilakarathna@sydney.edu.au

Albert Zomaya
University of Sydney
Sydney, Australia
albert.zomaya@sydney.edu.au

Jaybie Agullo de Guzman
University of the Philippines Diliman
Quezon City, Philippines
jaybie.de.guzman@eee.upd.edu.ph

ABSTRACT

Environmental understanding capability of augmented (AR) and mixed reality (MR) applications and devices are continuously improving through advances in sensing, computer vision, and machine learning. Various AR/MR applications have been developed that demonstrate such capabilities: i.e. scanning a space using a handheld or head mounted device and capturing a digital representation of the space that are undeniably accurate copies of the real space. However, these capabilities impose privacy risks to users: personally identifiable information can leak from captured 3D maps of the sensitive spaces and/or captured sensitive objects within the mapped space. Thus, in this work, we demonstrate how we can leverage 3D object regeneration for preserving privacy of 3D point clouds. That is, we employ an intermediary layer of protection to transform the 3D point cloud before providing it to the third-party applications. Specifically, we use an existing adversarial autoencoder to generate copies of 3D objects where the likeness of the copies from the original can be varied. To test the viability and performance of this method as a privacy preserving mechanism, we use a 3D classifier to classify and identify these transformed point clouds: i.e. perform super-class and intra-class classification. To measure the performance of the proposed privacy framework, we define privacy, $\Pi \in [0, 1]$, and utility metrics, $\mathcal{Q} \in [0, 1]$, which are desired to be maximized. Experimental evaluation shows that the proposed privacy framework can indeed varyably effect the privacy of a 3D object by varying the privilege level $l \in [0, 1]$: i.e. if a low $l < 0.17$ is maintained, $\Pi_1, \Pi_2 > 0.4$ is ensured where $\Pi_1, \Pi_2$ are the super- and intra-class privacy. Lastly, the proposed privacy framework can ensure relatively high intra-class privacy and utility, i.e. $\Pi_2 > 0.63$ and $\mathcal{Q} > 0.70$, if the privilege level is kept within the range of $0.17 < l < 0.25$.

CCS CONCEPTS

• Human-centered computing → Mixed / augmented reality;
• Security and privacy → Information accountability and usage control; Privacy protections.

KEYWORDS

mixed or augmented reality, privacy, 3D point clouds, point cloud classification, and adversarial autoencoders

1 INTRODUCTION

Advanced camera systems capable of mapping 3D spatial information is increasingly becoming popular among consumer devices.1 These improved camera systems support the recent developments in mixed reality (MR) technology particularly in improving environmental understanding capability: i.e. scan a space using a handheld or head mounted device and, then, capture a digital representation or mapping (i.e. spatial maps or 3D point clouds) of the space that are undeniably accurate copies of the real space. This enables MR applications to deliver realistic MR experiences, i.e. rendering virtual augmentations that seemingly inhabit the real and physical space.2

1For example, Apple Inc. has released the iPad Pro in 2020 with LiDAR, and subsequent iPhone releases in 2021 also has LiDAR.
2AR/MR SDKs of popular consumer platforms: Apple ARKit (https://developer.apple.com/documentation/arkit), Google ARCore (https://developers.google.com/ar/), and Windows MR API (https://developer.microsoft.com/en-us/windows/mixed-reality).
Concurrently, 3D object detection and classification have also become highly accurate and ubiquitous [27, 28]. While training and running these deep learning models is still compute and resource intensive, it is only a matter of time when such models can be run on mobile devices to be leveraged for environmental understanding and, hence, deliver better MR experiences. Nonetheless, the relatively memory light 3D point clouds can be sent to and processed remotely, i.e. on the edge or cloud, where deep learning models can be run more efficiently.

While these advanced environmental understanding empowers app developers to design innovative MR service, the access to accurate 3D mapping imposes unprecedented risks to user privacy [6, 16]. With the current access controls in place with these MR devices, regardless of the processing strategy—whether local or remote—the raw 3D point clouds are freely accessed by third party applications. As such, adversarial MR applications can easily access and collect 3D spatial maps which may include sensitive information: whether the captured space itself is sensitive, or portions of the captured space is sensitive. Aside from objects being detected, geometric and photo-metric characteristics of the objects, semantic relationships between objects, etc. may eventually be leveraged by adversaries which can potentially reveal personally identifiable information: whether about the device user, or the user’s association, linkage, or relationship to these spaces, objects, or other users.

Furthermore, the problem of user privacy is exacerbated by the fact that users do not have any control over the visual, and, now, spatial information shared to third parties. Due to binary user permission structure on current devices, the user can either share raw visual and spatial data or completely block permission to the camera. Not just users, device manufactures are also suffering from the lack of finer granular controlled release of spatial data, e.g. Oculus has completely blocked access to 3D point cloud data in order to mitigate unwanted data leakage. While this is good for user privacy, it significantly limits innovation and hinders the true potential of hardware capability.

In light of that, we propose a privacy-aware 3D spatial data sharing framework providing the user with the ability to control the level of privacy protection required. We introduce a 3D object regeneration mechanism as an intermediary layer of protection before sharing the 3D point cloud to third party applications as shown in Fig. 1. For the 3D point cloud regeneration, we employ a Hypernetwork-based Adversarial Autoencoder (AAE) and vary the level of likeness of the generated 3D object point cloud from the original. The level of likeness is obtained from the privilege level assigned by the user (cf. privilege slider in Fig. 1). That is, a higher likeness means a higher privilege is afforded to third-party applications, and lower, otherwise. The challenge is to release minimal amount of uniquely identifiable information regarding the 3D object, while also providing enough spatial information to allow the application to function with high quality of service. We first formulate adversarial models, as well as privacy and utility metrics. We then evaluate the viability of the proposed user configurable 3D regeneration approach as a privacy mechanism using a large 3D object database.

In this paper, we present the following contributions:

(1) Formalize the 3D privacy problem against a two-fold attacker which tries to reveal the super-class and intra-class labels of a 3D object; hence, we correspondingly define two privacy metrics, i.e. $\Pi_1 \in [0,1]$ for super-class, and $\Pi_2 \in [0,1]$ for intra-class.

(2) Define the utility metrics on which the privacy frameworks’ output will be measured against. We present two utility metrics based on two common MR functionalities: bounding box utility $Q_1$, and plane anchoring utility $Q_2$.

(3) Leverage existing 3D adversarial autoencoders (AAE) as an intermediary privacy-preserving mechanism for reducing 3D privacy leakage from 3D point clouds before exposing or sharing to third party applications.

(4) Evaluated the viability of the AAE-based privacy-preserving 3D regeneration using existing 3D deep neural network classifier to classify and identify the regenerated point clouds at various attacker privilege scenarios.

(5) Experimental evaluation shows that the proposed privacy framework can indeed variably affect the privacy of a 3D object by varying the privilege level $l$. For example, if low privilege level $l < 0.17$ is maintained, both $\Pi_1, \Pi_2 > 0.4$ is ensured.

(6) Lastly, the proposed privacy framework can ensure relatively high intra-class privacy and utility, i.e. $\Pi_2 > 0.63$ and $Q > 0.70$, at a privilege level range of $0.17 < l < 0.25$.

Overview. The rest of the paper is organized as follows. §2 presents the related work with emphasis and focus on recent 3D privacy work. §3 formalizes the 3D privacy problem followed by the formulation of the proposed 3D privacy framework, i.e. privacy-aware 3D point cloud regeneration, in §4. The experimental setup is detailed in §5, while the results are discussed in §6. We discuss the further implications of our work in §7. Finally, we conclude the paper in §8. The appendix shows supplementary plots and figures.

2 RELATED WORK

In this section, we present a comprehensive exposition of the recent related work aimed at privacy for 3D data, particularly related to MR. Then, we present a breadth of relevant 3D classification approaches followed by the adversarial attacks towards these classifiers.

2.1 3D privacy

While most of the earlier work in MR privacy were primarily aimed at visual (i.e. image and video) data protection [8], a handful of recent work have started to point out the potential privacy risks with the ubiquity of 3D “scanning” devices [3, 14, 30], and highlighted the ease-of-access of third-party applications and services over these 3D point cloud or spatial data [7, 12, 18].

3D leakage and further inference. On the MR-specific case, the actual risks brought about by indefinite access to spatial data captured by mobile MR devices has been looked into. Specifically, on how adversaries can easily identify spaces from captured 3D point cloud data (using Microsoft Hololens) [6, 9]. Other works have employed machine learning to reveal original scenes from 3D point cloud data by leveraging on additional visual information, i.e. SIFT (scale-invariant feature transforms) and other photometric information [26]. Subsequent work focused on a privacy-preserving method of pose estimation to counter the scene revelation [15, 34]: instead
of using 3D point cloud data, 3D line clouds are used during pose estimation to obfuscate 3D structures. The same line cloud approach has also been leveraged to translate the features to affine space to ensure that the features are privacy-preserving [11]. Another work also followed the same strategy in delivering privacy-preserving SLAM (synchronous localization and mapping) [33]. However, it has also been demonstrated that line clouds are still not immune from further inference as dense line clouds can still reveal 3D structures which allows for original scenes to be approximately revealed [5]. Moreover, the 3D line cloud approach only addresses specific functionalities such as pose-estimation and does not present its viability for surface or object detection which may be necessary for delivering immersive experiences in MR (e.g. to detect object on which virtual objects can be rendered or "anchored" onto). Thus, with current MR processing techniques, it would still be necessary for 3D point cloud data to be exposed but, perhaps, with some privacy-preserving transformations to hide sensitive content and prevent further spatial recognition.

3D encryption. An earlier approach proposed the encryption of 3D point clouds using chaotic mapping to conserve utility for view point histograms [19]. However, this method is reliant on sharing of keys, and once the keys— and, hence, access— are provided, no further privacy protection is ensured. Furthermore, this 3D encryption method was not tested against standard encryption approaches which have been employed in secure multi-party computation of 3D user models from encrypted user physiometric information for virtual clothe try-on [32]. Nonetheless, this 3D encryption approach together with other standard encryption methods (e.g. AES, and RSA) are essential part in providing protection along the entire MR service pipeline.

2.2 3D classification and reconstruction

2.2.1 3D classification. Now, as we explore risks associated with exposed 3D spatial data, we looked into the various methods that are utilized for 3D shape analysis, classification, and recognition. Early approaches employ description and inference by, first, computing 3D features and, then, utilizing these features for inference, say, by matching: examples include 3D Shape histograms [2], parametrized shape distributions [24, 25], vector field similarity [10], shape histograms [20, 21], curvature self-similarity [17], or heat-kernels [36]. On the other hand, the latest 3D recognition approaches have employed machine learning and has been more successful in 3D classification as well as segmentation. PointNet was the first to use deep learning, i.e. deep neural network (DNN), for 3D classification with point clouds as input [27]. PointNet++ presented some improvements in the underlying DNN architecture [28]. Latest developments on neural network-based 3D classification include the following: identifying salient regions of objects that inform about their class [41], improving robustness to rotations [22], leveraging alternating 3D projections for point cloud alignment or registration [29], and improving 3D segmentation processing time through point dropping [39].

2.2.2 3D adversarial approaches: attacks and regeneration. Unsurprisingly, 3D neural networks (NNs) are also susceptible to 3D adversarial point clouds which can confuse 3D NNs. Recent efforts focused on demonstrating 2D adversarial approaches to 3D [23, 37], and providing defenses against such adversarial attacks [40, 42]. These efforts are motivated by mission-critical 3D information which requires that 3D NNs be more resilient to adversarial attacks when classifying 3D objects, e.g. detecting pedestrians for self-driving cars.

However, we argue that for most non-critical 3D applications, users may desire to have less of their environments be recognizable to machines (and, hence, potential adversaries). Adversarial autoencoders (AAEs) can be used to transform and generate 3D point clouds to potentially preserve privacy. Recent works focused on improving AAE using hypernetworks [35], or using curvature functions with canonical shapes as reference [38]. However, none of these methods are originally designed for privacy-preservation, but can be leveraged and modified to be utilized as such as we propose in this work.

3 PROBLEM FORMULATION - 3D OBJECT PRIVACY IN MIXED REALITY

In the following sections, we introduce the representation of 3D spatial data used in our analysis, formulate the adversary models, establish the intended utilities of the MR application, and define the privacy metrics that we consolidate in our proposed solution.

3.1 3D spatial data

Current commodity MR devices use one or combination of SLAM (simultaneous localization and mapping), SFM (Structure from motion) or visual odometry to capture and represent its surrounding 3D environment [31]. This 3D structure representation usually comes in the form of a point cloud which contains a set of unordered and permutationally invariant points, each representing a point, i.e. an edge or part of a surface, in space (in terms of x-y-z coordinates) with no relationship defined among the points. Although the mapped 3D surroundings or objects might subsequently be converted to different representations such as 3D voxel grids, meshes, or oriented point clouds, the 3D point clouds which is often the direct output of the captures still encode substantial amount of data, and can even be directly used for processing. In this paper, we conduct our analysis on 3D objects while representing them as unoriented point clouds (with no normal vector attached).

3.2 Adversary models

To formally define the problem and the privacy and utility metrics used in this work, we develop upon the formalization introduced in [9] and [6]. We introduce $S$, a representation of an object captured from an MR user’s surrounding as the key entity of the 3D object privacy problem. We identify $S$ with two key attributes; it belongs to a super-class object category based on the practicality (e.g. chair, and table), and it represents a unique member within its super-class in terms of 3D shape, structure, make, brand, etc. (e.g. product code 604.169.25 from IKEA 2021 catalog).

The MR application offers a functionality $G$, in which the object representation $S$ is processed in some way to generate the output $Y$. The utility of the 3D object $Q$ gives a measurement of the quality of the produced $Y$ which is required for the functionality $G$ to be derived. While an adversarial attacker formulates a hypothesis $h$ on a certain attribute of $S$, the privacy preserving regeneration mechanism generates a privacy-preserved version $\tilde{S}$ corresponding to input $S$, expecting to reduce the accuracy of $h$ formulated by
the attacker. For the convenience of the reader, we include all the notations used in our work in Tab. 1.

Formalizing the attacker. In this paper, our main focus is on object inference attacks. More specifically, the adversarial attacker that we present in this work attempts to identify a 3D object represented by a point cloud in two main hierarchical levels: (1) Super-class inference, i.e. identifying the high-level category of the object according to its practicality; and (2) Intra-class inference, i.e. identifying the unique object within the super-class. For each scenario, the attacker attempts to narrow down the assumed super-class/intra-class to a subset of previously encountered references.

Let $S$ be a known point cloud within the reference set where $k_i$ is the object’s super-class label (i.e. $k_i \in K$ where $K$ is the set of all the super-class object labels), and $m_{j|i}$ is its intra-class label within super-class $k_i$ (i.e. $m_{j|i} \in M_i$ and $M_i$ is the set of all the intra-class object labels of super-class $k_i$). Note that $k_i, m_{j|i} \in \mathbb{Z}^+$. Now, the attacker makes two-fold hypothesis about a query point cloud $S'$ with unknown super-class label $k'_i$ and intra-class label $m'_{j|i}$, i.e. we denote with $\ast$ the unknown labels to the attacker:

\text{Attacker} : S' \rightarrow h_{1,p_1}, h_{2,p_2}

Super-class hypothesis- $h_1$. The attacker makes a super-class hypothesis $h_{1,p_1}$ that the unknown super-class label $k'_i$ is within a “basket” subset $Y'_k \subset K$ of super-class labels:

$$h_{1,p_1}: k'_i \in Y'_k = \{k_1, k_2, \ldots\}, \text{ where } p_1 = \frac{|Y'_k|}{|K|}$$

The parameter $p_1$ is the ratio of the size of the super-class hypothesis basket $Y'_k$ to that of the known super-class set $K$. When $p_1|K| = 1$, i.e. $Y'_k = \{k_x\}$, $k_x$ is the top-1 super-class hypothesis.

Intra-class hypothesis- $h_2$. Next, the attacker makes an intra-class hypothesis $h_{2,p_2}$ given the $h_1$ super-class label $k_x \in K$, that the unknown intra-class label $m^*_{j|x}$ is within a set $\eta^*_m \in M_k$ of intra-class labels:

$$h_{2,p_2}: m^*_{j|x} \in \eta^*_m = \{m_1|x, m_2|x, \ldots\}, \text{ where } p_2 = \frac{|\eta^*_m|}{|M_k|}$$

Similarly, $p_2$ is the ratio of the size of the intra-class hypothesis set $\eta^*_m$ of the known intra-class set $M_k$ of super-class $k_x$.

As $\rho$ increases, the attacker will identify the query $S'$ as a member of a larger set of possible super-classes and unique-objects. Therefore as $\rho$ increases, the attacker’s ability to precisely reidentify the object decreases, hence the privacy threat that the attacker can pose to the user decreases. Hence, an attacker will ideally try to narrow down the hypothesis to a subset with as small $\rho$ as possible.

### 3.3 MR application utility

Various functionalities are offered by MR applications that leverage 3D data from the user environments. Such functionalities vary in terms of level of details required in the 3D scans to effectively deliver the intended functionality. A given functionality $G_o$ requires a particular output $Y_o$ from the device obtained by applying some operation on the raw point cloud scans: i.e. $G_o(S) \rightarrow Y_o$. The privacy mechanism takes original point cloud $S$ as input, and generates a privacy-preserved version $\tilde{S}$ of it. Thus, we can define a utility metric for the regenerated point cloud with respect to the original point cloud by using an output transformation error for the particular application functionality, where $Er_o$ is a distance/error function that quantifies the dissimilarity between $Y_o$ and $Y_{o,\tilde{S}}$: i.e. $G_o : Er_o(\{Y_o, Y_{o,\tilde{S}}\}) \rightarrow Q_{0,\tilde{S}}$.

In this work, we analyze two main MR application utilities that require varying levels of details in the output $Y_o$, and we define utility metrics separately for each.

Bounding box utility. The most basic and naive MR functionality requires the general location of an object, often to be treated as an obstacle in the 3D environment. We define this functionality as $G_1$ whereas the output $Y_{1,S}$ represents the general location and size of the object represented by $S$ in the form of a 3D bounding box: i.e. $G_1 \rightarrow Y_{1,S} = (x_S, y_S, z_S, x_0, y_0, z_0)$, where $x, y, z$ indicate the dimensions of the bounding box in 3 axes, and $x_0, y_0, z_0$ indicate the position of the center of the bounding box.

Next, we formulate the utility metric for the MR functionality $G_1$. Here, we can skip the intermediate error function and directly formulate the utility metric to quantify the similarity of object bounding boxes in terms of 3D Intersection-over-union (IoU).

$$Q_{1,S,S} = IoU(Y_{1,S}, Y_{o,\tilde{S}})$$

Plane anchoring utility. The second and most widely used MR functionality $G_2$ involves anchoring 3D virtual objects onto 2D surfaces of real objects. For this, we define the function output: i.e. $G_2 \rightarrow \{Y_{2,S} : p, Y_{2,\tilde{S}} : \tilde{p}\}$, where $p$ and $\tilde{p}$ indicate the point clouds representing the most prominent 2D plane of the object from the original and regenerated point clouds separately.

3D object anchoring in MR could be either static or dynamic. In static anchoring, anchored object is stationary, and MR functionality only requires the position and normal vector for one point on the plane. Hence, we formulate the transformation error in terms of the angle and distance between the planes $p$ and $\tilde{p}$. For dynamic anchoring, anchored object moves around the plane, therefore functionality requires the area and the point distribution of the 2D plane.

### Table 1: Table of notations

| Notation | Description |
|----------|-------------|
| $S$      | Point cloud |
| $k_i$    | $i^{th}$ super-class label |
| $m_{j|i}$| $j^{th}$ intra-class label given a super class $k_i$ |
| $l$      | variable privilege level, i.e. $l \in [0, 1]$ |
| $S'$     | Query point cloud with unknown labels |
| $\mathcal{S}$ | Regenerated Point cloud |
| $h_1(S')$| hypothesis 1 for query $S'$ |
| $h_2(S')$| hypothesis 2 for $S'$, given super-class $k$ |
| $Y'_k$   | super-class label subsets formed by hypothesis 1 |
| $\eta^*_m$| intra-class label subsets formed by hypothesis 2 |
| $p_1, p_2$| attack score functions for hypothesis set size |
| $\mathcal{L}_{h_1(S')}|\eta^*_m|$ | Likelihood function of hypothesis 1 |
| $\mathcal{L}_{h_2(S')}|\eta^*_m|$ | Likelihood function of hypothesis 2 |
| $\Pi_1(S'), \Pi_2(S')$ | super- and intra-class privacy metrics for query $S'$ |
| $G_o(S)$ | $\rho^H$ MR functionality for point cloud $S$ |
| $Y_o(S)$ | output of $\rho^H$ MR functionality for input $S$ |
| $Er_{o,\tilde{S}}$ | Transformation error for $\rho^H$ utility |
| $Q_{0,\tilde{S}}$ | $\rho^H$ utility metric for regenerated $\tilde{S}$ w.r.t $S$ |
Hence, we additionally use difference in area, and difference in point distributions between $p$ and $\bar{p}$ to define the error function.

$$
\begin{align*}
E_{2: Static} &= 0.5|1 - \tilde{n}_p \cdot \tilde{n}_p| + 0.5|\tilde{r}_p - \tilde{r}_p| \\
E_{2: Dynamic} &= 0.25|1 - \tilde{n}_p \cdot \tilde{n}_p| + 0.25|\tilde{r}_p - \tilde{r}_p| \\
\end{align*}
\tag{4}
$$

Here, $\tilde{n}_p$ indicates the unit normal vector to the 2D plane represented by $p$; thus, $|1 - \tilde{n}_p \cdot \tilde{n}_p|$ calculates the angle between the planes. $\tilde{r}_p$ indicates the perpendicular vector from origin to the point on plane $p$; thus, $|\tilde{r}_p - \tilde{r}_p|$ measures the distance between planes. $A(p)$ measures the area of the 2D convex-hull formed by $p$, and $CD(p, \bar{p})$ returns the Chamfer distance between two point clouds $p$ and $\bar{p}$, reflecting the point distribution discrepancy between them.

Before deriving the utility function $Q_{2: Static}$ from $E_2$, we separately apply min-max normalization for each component $|1 - \tilde{n}_p \cdot \tilde{n}_p|$, $|\tilde{r}_p - \tilde{r}_p|$, $|A(p) - A(\bar{p})|$, $CD(p, \bar{p})$

The modified error functions obtained after the said normalization are as follows: $E_{2: Static} \rightarrow E'_{2: Static}$ and $E_{2: Dynamic} \rightarrow E'_{2: Dynamic}$. Finally, we formulate the second (twofold) utility metric based on the functionality $G_2$:

$$
\begin{align*}
Q_{2: Static} &= 1 - E'_{2: Static} \\
Q_{2: Dynamic} &= 1 - E'_{2: Dynamic} \\
\end{align*}
\tag{5}
$$

### 3.4 Privacy metrics

We characterize the privacy introduced by the proposed mechanism for a particular 3D point cloud $S$ with respect to the type of adversarial inference attempted upon by the attacker: (1) Super-class privacy, and (2) Intra-class privacy. We formulate the privacy metrics for both of the above scenarios using the expected error in hypotheses $h_1$ and $h_2$. Furthermore, since the hypotheses are given as sets of labels, i.e., $\gamma^*_k$ and $\eta^*_m$, the privacy metrics are also dependent on the size of these label sets which are represented as fractions of the reference sets by the parameters $\rho_1$ and $\rho_2$.

#### 3.4.1 Super-class privacy.

In trying to infer a query point cloud $S^*$, the attacker generates a likelihood function for $\gamma^*_k$:

$$
L_{h_1(S^*)}(\gamma^*_k) = \sum_{\forall k \in \gamma^*_k} \sigma_1(k, S^*)
$$

$\sigma_1(k, S^*)$ is an attacker super-class score function for a hypothesis super-class label $k$ and for the unknown point cloud $S^*$, i.e., it represents the inclination of the attacker to match the query $S^*$ with the particular super-class, such that those values for each super-class adds up to 1:

$$
\sigma_1(k, S^*) \leq 1, \sum_{\forall k \in \gamma^*_k} \sigma_1(k, S^*) = 1
\tag{6}
$$

Then, the super-class privacy $\Pi_1$ for a query point cloud $S^*$ with actual super-class label $k_l$ can be expressed in terms of the expected error in hypothesis set $\gamma^*_k$:

$$
\Pi_1(S^*) = \delta(k_l, \gamma^*_k) L_{h_1(S^*)}(\gamma^*_k) + (1 - \delta(k_l, \gamma^*_k))(1 - L_{h_1(S^*)}(\gamma^*_k))
\tag{7}
$$

where $\delta(s, S)$ is a set membership function, i.e.,

$$
\delta(s, S) = \{1 \text{ if } s \in S; 0 \text{ if } s \notin S\}
$$

### 3.4.2 Intra-class privacy.

Next, given a particular hypothesis super-class label $k$, the attacker generates a likelihood function for the intra-class hypothesis label set $\eta^*_m$:

$$
\bar{L}_{h_2(S^*)}(\eta^*_m) = L_{h_2(S^*)}(k) \cdot L_{h_2(S^*)}(\eta^*_m)
$$

where $L_{h_2(S^*)}(\eta^*_m) = \sum_{\forall m \in \eta^*_m} \sigma_2(m, S^*)$

$\sigma_2(m, S^*)$ is an attacker intra-class score function for a hypothesis object label $m$ and for the unknown point cloud $S^*$ within a given super-class $k$, i.e., it represents the inclination of the attacker to match the query $S^*$ with the particular object $m$, given that $S^*$ belongs to the super-class $k$, such that those values for each object within $k$ adds up to 1:

$$
\sigma_2(m, S^*) \leq 1, \sum_{\forall m \in \eta^*_m} \sigma_2(m, S^*) = 1.
\tag{8}
$$

The intra-class privacy $\Pi_2$ for a query point cloud $S^*$ with actual super-class label $k_l$ and intra-class label $m_{ji}$ is also calculated as the expected error of the attacker hypothesis where the error of the super-class hypothesis affects the intra-class hypothesis:

$$
\Pi_2(S^*) = \sum_{\forall k \in \gamma^*_k} \bar{L}_{h_2(S^*)}(k) + \bar{L}_{h_2(S^*)}(k_l)(1 - \delta(m_{ji}, \eta^*_m))L_{h_2(S^*)}(\eta^*_m) + (1 - \delta(m_{ji}, \eta^*_m))(1 - L_{h_2(S^*)}(\eta^*_m))
\tag{9}
$$

We leverage the privacy metrics and the utility metrics formulated in this section to measure and evaluate the performance of the proposed framework in various contexts as given in §6.

### 4 3D PRIVACY FRAMEWORK

Our framework utilizes an Auto-Encoder based point cloud regeneration mechanism to introduce privacy preserving versions for point cloud objects.

#### 4.1 AAE-based point cloud regeneration

The deep neural network used in our framework is called HyperCloud [35] which consists of an Adversarial Auto-Encoder (AAE) and a target network. The AAE works as a hyper-network ($H$) for the target network ($T$) i.e. the AAE returns the weights of the target network as output. The AAE takes a point cloud as the input ($X$) which passes through an encoder ($E$) and a decoder ($D$) to return the weights ($\theta$) of the target network.

In addition to $E$ and $D$, the AAE also includes another neural network called discriminator. The goal of the discriminator is to learn to distinguish between the latent space ($E(X) \sim Z$) and the prior distribution ($P$) by using a regularization loss term in the cost function. The target network with the weights $\theta$ i.e. $T_\theta$ accepts a baseline distribution (3D ball or sphere) as input and generates a reconstruction ($Y$) of the original 3D point cloud. The cost function includes the regularizer term and the Chamfer pseudo-distance between original point cloud and the reconstruction as follows:

$$
cost(X; E, D, Y) = CD(X; Y) + Reg(E(X); P)
\tag{10}
$$

Where $CD$ refers to the Chamfer pseudo-distance and $Reg$ refers to the regularization loss term. The Chamfer pseudo-distance can alternatively be replaced with the Earth Mover’s distance. If $P$ is
We save the weights of the hypernetwork to create a weight bank (referring to lower epoch in the architecture training) are for higher privilege settings, and those with lower loss (usually referring to lower epoch in the architecture training) and more accurate reconstructions. We have used this property decreases, Chamfer pseudo-distance also decreases implying better the reconstruction using the architecture; so, as the loss function decreases, thereby minimizing the possible privacy risk. The slider amount of detail in the 3D data is limited by the proposed mechanism. When high privilege is given, the third party MR application can collect 3D information from the user’s surrounding with high level of detail, hence can cause larger privacy risk to the user. Similarly in the next step, we use the Softmax layer to define the scoring function in Eq. 6, \( \sigma_1(k; S^*) \) for each super-class \( k \) in the reference set. Then, the attacker formulates the hypothesis in Eq. 1 by solving the following optimization.

\[
h_1(p_1) = y^*_k = \underset{y^*_k}{\text{Argmax}} \sum_{k \in \mathcal{K}} \sigma_1(k, S^*)
\]

Similarly in the next step, we use the Softmax output of the intra-class classifier trained on a particular super-class \( k_x \) to formulate the scoring function in Eq. 8, \( \sigma_2(m; S^*|k_x) \) for each object \( m \) in the intra-class reference set of super-class \( k_x \). We formulate the hypothesis in Eq. 2 by solving the following optimization:

\[
h_2(p_2) = \eta^*_m = \underset{\eta^*_m}{\text{Argmax}} \sum_{m \in \mathcal{M}_x} \sigma_2(m, S^*|k_x)
\]

Note that the cardinalities of the subsets \( y^*_k \) and \( \eta^*_m \) obtained in Eq. 11 and Eq. 12 are parameterized by \( p_1 \) and \( p_2 \) as shown in Eq. 1 and Eq. 2. We evaluate the performance of our attackers with respect to these parameters in §6. Moreover, for this experimental setup, the super-class reference set size \( |K| = 10 \), and intra-class reference set size \( |M_k| = 100 \) for each \( k \in K \).
5.2.2 Attacker classification. Both super-class and inter-class adversarial inference is done with respect to a set of reference 3D objects that are previously encountered and learnt by the attacker in one or many of the following ways: from publicly available data, from the data intercepted previously from the same user or different users, or data obtained via other potentially colluding applications. In a situation where the query object is not present in the references (i.e. not previously encountered by the attacker), the attacker still attempts to identify the query as the structurally closest (most similar) reference object.

We define the following four types of attackers based on the approach of acquiring and the quality (in terms of the ‘exposed’ privilege level) of 3D data in their reference sets:

- **J1** - public-aware attacker - Using publicly available data as a set of reference 3D objects
- **J2** - low-privilege-aware attacker - Intercepting data from users with low privilege settings
- **J3** - medium-privilege-aware attacker - Intercepting data from users with medium privilege settings
- **J4** - high-privilege-aware attacker - Intercepting data from users with high privilege settings

In §6.2, we present our analysis on how the proposed framework effectively preserve privacy against these different attackers.

5.2.3 Simulating attacker learning process. We created the reference dataset for the attacker J1 by using publicly available dataset as mentioned in §5.1. We created 100 normalized augmentations for each unique object in all the super-classes by adding random rotations and Gaussian noise to complete the reference (training) dataset. For attackers J2, J3 and J4, we used our privacy framework to create 100 regenerations for each unique object in every super-class at various privilege levels (allowed for the attacker) to create the reference (training) datasets.

As explained in §4.2, the training epoch number, which the hypernetwork weights are loaded from, has a direct proportional relationship with the provided privilege level. Therefore, we divided range of I to three abstract privilege settings based on the epoch ranges. The privacy framework based on Hypercloud was trained for 300 epochs in this experimental setup:

- Low privilege (0-0.167) - epoch range (0,50]
- Medium privilege (0.167-0.232) - epoch range [50,70]
- High privilege (0.232-1) - epoch range [70,300]

This division was done based on experimental results that we obtained, as will be explained in §6. For example, for the attacker J2, we randomly chose 100 epoch numbers within the range (0,50] with replacement and regenerated each unique object m ∈ Mk, for every super-class ∀k ∈ K, using the hypernetwork loaded with weights

![Figure 4: Reference (training) dataset generation for the super-class “Chair”](image)

![Figure 5: Classifier training process for an Attacker J: 1x super-class classifier, and 10x intra-class classifiers (i.e. one for each super-class). Each attacker generates their own set of classifiers.](image)

![Figure 6: Testset generation for the super-class “Chair”](image)

from that particular epoch number. A similar procedure was followed for the attackers J3 and J4, except the epoch ranges were changed to [50,70] and [70,300] respectively to reflect each privilege setting. The reference dataset generation for a single super-class (e.g. “Chair”) is shown in the Fig. 4.

Now, for each attacker, we have a reference dataset of size 10 × 100 × 100, i.e. 100 reconstructions of each of the 100 unique objects for all the 10 super-classes. The attacker uses this reference data for training and validation of their classification models, and uses them to infer the super-class and intra-class attributes of a query point cloud S∗.

For super-class classification, each attacker trains one classification model which uses the labeled dataset of size 10 × 100 × 100 where we have 100 × 100 samples for each super-class. For intra-class classification, each attacker trains 10 different classification models, i.e. one for each super-class. Each of these 10 models uses the dataset of size 100 × 100, where we have 100 labeled samples (regenerations) for each unique object. An attacker’s classifier training process is shown in Fig. 5.

5.2.4 Evaluating the attacker performance. To assess each attacker’s success, we test all of their classification models against a scenario where it attempts an object reidentification attack (following the attacker hypothesis formulation in §5.2.1) while using the proposed privacy framework. As discussed in §4.2, according to the user input on the privilege slider, we choose its corresponding epoch number from within the range (0,300] and load the corresponding weights into our framework. This network is used to create a regeneration of the original point cloud with the privilege allowed by the user.

For each of the 100 unique objects from all the 10 super-classes, we created 3 reconstructions at every other epoch, i.e. all the even numbered epochs in the range (0,300]. This labeled dataset of size 10 × 100 × 150 × 3 was used to test all the 11 classifiers trained by each attacker. Our testing dataset will allow us to assess the attacker’s success at inferring both the super-class and the intra-class attribute at various levels of privilege set by a user. The testing dataset generation for a single super-class is shown in Fig. 6.
We obtain a set of points where virtual objects are most likely to be anchored on. But the
utility metric for the top-1 success of Attacker $J_1$ for selected 3D objects: (a) chair, (b) table, (c) bed, and (d) bench.

5.3 Utility evaluation for regenerated point clouds

Higher privacy usually comes at a cost of quality of service. The goal is for the privacy framework to still allow MR applications to successfully deliver their intended functionalities while preserving the privacy of the point cloud objects. To this end, we utilize two utility metrics defined in §3.3, along with an additional Chamfer distance measurement to evaluate the regenerated point clouds.

In order to realize the utility metric $Q_{LS,S}$ from Eq. 3, we calculated the axis-aligned bounding boxes for each original point cloud $S$ and compared it with the bounding boxes of regenerated point clouds $\bar{S}$ from each privilege level $l$. We use the standard 3D Intersection-over-union - IoU as the comparison metric in $Q_1$.

Next, to realize the utility metric $Q_{SS,S}$ from Eq. 5, we used a modified implementation of RANSAC [13] algorithm to extract the most prominent 2D plane of each point cloud object. Since our dataset consists primarily of household objects, we conducted our analysis for 2D horizontal planes, e.g., sitting plane of chairs where virtual objects are most likely to be anchored on. But the utility metric $Q_{SS,S}$ can be applied to any type of plane of choice. We obtain a set of points $P$ that are fitted to the most prominent 2D plane of the object by the RANSAC algorithm, and calculate the metrics given in the Eq. 5 separately for static and dynamic anchoring cases.

Additionally, some MR functionalities might require near-truth representation of the complete structure of the object. Therefore its also important to investigate how the object structure of $S$ is preserved in privacy-preserving regenerations $\bar{S}$ created from $S$. To this end, we calculate the well known Chamfer distance between the original point cloud $S$, and its regenerated point clouds $\bar{S}$ at each $l$ privilege level. According to Eq. 10, Chamfer distance is also a part of the cost function in our framework. It essentially measures the squared distance between each point in one point cloud to its nearest neighbor in the other point cloud.

6 RESULTS AND DISCUSSION

6.1 Privacy evaluation for the top-1 hypothesis of Attacker $J_1$

We evaluated the performance of our proposed privacy-preserving regeneration separately for each super-class. In Fig. 7, x-axis indicates the privilege level $l$ at which a query point cloud has been regenerated. Each plot indicate the variation of the two privacy metrics $\Pi_{1,B}$ and $\Pi_{2,B}$ are used as baselines to evaluate the performance increase of the proposed framework in terms of privacy preservation against Attacker $J_1$.

In Fig. 7 and Fig. 8, the baselines calculated in §5.2.5 are shown for the proposed framework. As $l$ increases, the regeneration becomes increasingly similar to the original point cloud: eventually decreasing the privacy for both super-class and intra-class attack scenarios. One notable observation is that the privacy for a particular regeneration is higher for the intra-class case than for the super-class case. This is due to the penalization in Eq. 9 for failed super-class hypothesis in the first step of hypothesis 2 formulation.

In Fig. 7 and Fig. 8, the baselines calculated in §5.2.5 are shown in dashed lines. The privacy metric $\Pi_2$ for the proposed framework is significantly large in comparison to baseline values in all shown super-classes, while privacy metric for $\Pi_1$ is large in comparison to baselines for relatively low $l$ values, and gets closer to the baseline value for higher $l$ values. For example, according to Fig. 8, proposed framework adds 0.58 increase in $\Pi_1$ privacy, and 0.81 increase in $\Pi_2$ privacy at privilege level $l = 0.167$, and respectively 0.23 and 0.62 increase in $\Pi_1$ and $\Pi_2$ at $l = 0.232$.

A significant change in both $\Pi_1$ and $\Pi_2$ can be observed within the approximate range of $l = 0.167$ and $l = 0.4$ followed by a slow decay thereafter, indicating the potential of the proposed regeneration mechanism to be used as a tunable privacy encoding for point cloud objects.
6.2 Privacy performance against various attackers with different privilege levels

In §5.2, we introduced 4 different attackers based on how each attacker obtain the reference data for its hypothesis formulation. Out of them, Attackers $J_2, J_3$, and $J_4$ can be analyzed separately since they use regenerated point clouds to form its reference set. This resembles a practical situation where a user is already using the proposed framework to transform the 3D objects prior to being released to the MR application, an adversarial attacker can still intercept these transformed data and learn to formulate hypotheses. In Fig. 9, we present the analysis of the variation of privacy metrics for different Attackers $J_2, J_3$ and $J_4$. The results are aggregated for all super-classes and all unique-objects within each super-class.

One notable observation is the clear difference of privacy metric values for different attackers. The regenerated point clouds show very high privacy with respect to Attacker $J_2$, which is trained only on low privilege data, while showing significantly low privacy metric values for Attacker $J_4$, which is trained on high privilege data. Similar to the results in §6.1, intra-class privacy metric values are comparatively higher than that of super-class for all three attackers. On the other hand, the privacy of regenerated point clouds with respect to Attacker $J_4$ is even lower than that for Attacker $J_1$ that we discussed in §6.1. The potential reason is because the features learnt by Attacker $J_4$ from the regenerated point clouds with highest privilege are more similar to the query point clouds than the features of original point clouds (from public datasets). The dip in the plots for Attackers $J_2$ and $J_3$ indicate that the attackers can pose a higher privacy threat for point clouds which are regenerated in the similar $l$ range as their reference sets, but the threat does not generalize well for other $l$ ranges: e.g. trained at lower $l$ but does not perform well against query objects released at higher $l$.

Given the difference in how Attacker $J_1$ and the rest of the attackers ($J_2, J_3$ and $J_4$) learn their reference sets, two types of privacy threats can be identified. Since Attackers $J_2, J_3$, and $J_4$ are only exposed to regenerated point clouds from the proposed framework, they lack knowledge about the unique objective features of the original point cloud object. The inference of a query point cloud $S^*$ by such attackers can be identified as an object reidentification attack, where the query can be matched with one of its reference super-classes/objects, but the attacker cannot derive further information or pose additional threats since the regenerated objects do not necessarily resemble their original counterparts. On the other hand, attacker $J_1$ learns its reference set from publicly available data, where additional object, semantic, and contextual features are also possibly available for a given object. Once Attacker $J_1$ infers a query point cloud $S^*$ with respect to such a reference set, in addition to an object reidentification threat, revealing the true identity of the object can lead to multiple other privacy threats. Although not reflected in the privacy metrics $\Pi_1$ and $\Pi_2$ in Fig. 9, it is noteworthy that $J_1$ has comparatively larger potential to pose additional privacy threats.

6.3 Privacy metric evaluation for varying sizes of hypothesis sets

In Eq. 1 and Eq. 2, we introduced the hypothesis formulation in terms of the parameters $\rho_1$ and $\rho_2$. In simple terms, given a query point cloud $S^*$, the attacker attempts to narrow down the possible identity of it to a portion of its reference set, and $\rho$ parameterizes the size of the hypothesized subset as a portion of the reference set. Intuitively, when $\rho$ increases, the hypothesis subset size increases, hence its becomes increasingly probable that the query point cloud $S^*$’s identity is included in the hypothesis subset. This behaviour is reflected in privacy metrics $\Pi_1$ and $\Pi_2$ as shown in Fig. 10. But at the same time, for large $\rho$ values, it becomes increasingly harder for the attacker to pose a unique privacy threat to the user.

According to the violin plots in Fig. 10, for low privilege point clouds, the increase in $\rho_1$ drastically drops the $\Pi_1$ privacy, whereas for high privilege point clouds, $\Pi_2$ is already low, therefore the increase in $\rho_1$ only induces a small drop in $\Pi_2$ privacy. On the other hand $\Pi_2$ for low privilege point clouds is already ~ 1, therefore $\rho$ should be increased by a significant fraction to induce a decrease $\Pi_2$ privacy. Another notable behaviour is that the privacy metrics $\Pi_1$ or $\Pi_2$ vary non-linearly with $\rho$ in almost all different privilege regions.

6.4 Utility results

In this work, we leveraged two utility metrics based on two generic MR application functionalities that require 3D information in different levels of details as explained in §3.3. Fig. 11 shows the aggregated utility metrics values (averaged across 10 super-classes and 100 unique objects per each super-class) from the test dataset.
Additionally, we also use Chamfer distance as a measurement of structural dissimilarity between the original point cloud and its regenerated counterparts at different privilege levels. This measurement gives insight into the success of regenerated point clouds in delivering MR functionalities which require near-truth representations of objects.

Utility metric $Q_1$ given in Eq. 3 calculates the IoU between the bounding boxes that encompasses the original point cloud and the regenerated point cloud. A higher IoU implying that the dimensions and location of the regenerated point cloud is more similar to that of the original. As shown in Fig. 11a, $Q_1$ reaches 0.78 by 0.167 percentage privilege (low privilege), and 0.83 by 0.232 percentage privilege (medium privilege), and maintains a $Q_1$ of ~0.9 between percentage privileges 0.332 to 1 (high privilege). It is evident that the utility of the proposed regeneration mechanism with respect to $Q_1$ utility converges quickly, indicating a reasonably good utility guaranteed even for point clouds released with lower privileges.

Figures 11b-11c shows the aggregated results for the test dataset with respect to $Q_2$ given in Eq. 5. We implemented a RANSAC based plane fitting to find the point clouds corresponding to the most prominent 2D horizontal plane in each transformed object, and compared its attributes with the corresponding plane from the original object. Since our dataset is limited to household objects, and anchoring virtual objects predominantly happen for horizontal planes, we conducted the analysis for horizontal planes. By $l = 0.167$, $Q_2$:Static and $Q_2$:Dynamic reach 0.70 and 0.69 respectively, and by $l = 0.232$, $Q_2$:Static and $Q_2$:Dynamic reach 0.84 and 0.79.

The unexpected variations in the $Q_2$ utilities in the very low $l$ range, $l < 0.1$, can be accounted for the abrupt variation of point cloud re-generations with high loss values.

Next, Fig. 11d shows the Chamfer distance between original and regenerated point clouds (at varying privilege levels) averaged across all super-classes unique objects in each super-class. As shown in the graph, by 0.167 percentage privilege, the average chamfer distance has already decreased to 0.03, and by 0.232 percentage privilege, to 0.01. This indicates that the regenerated point clouds from the proposed framework are able to effectively retain the structural properties and its associated utilities even in the point clouds released with lesser privilege. Finally, the error margins in every utility metric in Fig. 11 makes it evident that the proposed framework allows the discussed utilities consistently across a large number of super-classes and objects despite the large variation in their shapes and structures.

6.5 Privacy-utility trade-off

Fig. 12 shows how the privacy metrics $\Pi_1$ and $\Pi_2$, and the utility metrics $Q_1$, $Q_2$:Static, and $Q_2$:Dynamic vary with different privilege levels ($l$) allowed by the privilege slider. This analysis is done over the test dataset explained in §5.1 using the Attacker $J_1$ which is trained on publicly available data, for hypothesis subset size=$1$. Consistent with the results from previous sections (§6.1, §6.2, and §6.4), the privacy metrics drop (with the super-class privacy dropping faster than the intra-class privacy) while the utility metrics increase drastically as the exposed privilege level $l$ is increased. Baseline privacy metrics $\Pi_1:B$ and $\Pi_2:B$ are shown in dashed lines for comparison.

At $l = 0.10$, a relatively high super-class privacy, i.e. $\Pi_1 > 0.63$, can only be maintained at low utility values, i.e. $0.50 < Q < 0.57$. On the other hand, the same relatively high intra-class privacy, i.e. $\Pi_2 > 0.63$, can be maintained while providing potentially higher utility, i.e. $0.83 < Q < 0.87$, at $l = 0.25$. For regenerations with $l \leq 0.25$, the lower bound of $\Pi_1$ is 0.09, and the lower bound of $\Pi_2$ is 0.63. It implies that for privilege levels $l \leq 0.25$, although there’s a high chance of the object’s super-class label to be exposed, it’s uniqueness within that super-class cannot be ascertained by an attacker.
In the medium privilege region (e.g., \( l = 0.20 \)), privacy metric values \( \Pi_1 = 0.21 \) (0.17 improvement from baseline) and \( \Pi_2 = 0.85 \) (0.68 improvement from baseline) can be guaranteed for a utility \( 0.77 < Q < 0.81 \). Moreover, we can draw a “goldilocks” privilege level range of \( 0.17 < l < 0.25 \) which allows us to really ensure both high intra-class privacy, i.e. \( \Pi_2 > 0.63 \), and utility, i.e. \( Q_1, Q_2 > 0.70 \).

**AUC as privacy mechanism efficiency.** Fig. 13 shows the variation of the privacy metrics vs. different utility metrics including the Chamfer distance. The objective of the proposed privacy framework is to regenerate point clouds so that the utility is maximized (transformation error minimized) while the privacy against object inference/reidentification attacks is also maximized. For the case of Chamfer distance, structural similarity of regenerations should be maximized by minimizing the Chamfer distance, while maximizing ensured privacy. Hence, we formulate a generic measure for the efficiency of any privacy mechanism in terms of the Area under the curve - AUC of the Privacy-Utility graph. Note that the AUC measurement for chamfer distance in Fig. 13d is normalized by the total area of the plot.

As shown in Table 2, using the AUC to measure the efficiency of our proposed mechanism, we can see that the AUC range is at the medium level with \( 0.49 < \text{AUC} < 0.64 \) for super-class privacy, while the AUC range is at the high level with \( 0.84 < \text{AUC} < 0.89 \) for intra-class privacy. This corroborates our earlier discussion that the proposed privacy framework can maintain high intra-class privacy while ensuring high utility despite that an equally high super-class privacy isn’t guaranteed. Again, the privilege level range of \( 0.17 < l < 0.25 \) ensures this performance as shown in Fig. 12.

### 7 LIMITATIONS & FUTURE WORK

Despite not being able to ensure the super-class privacy, the proposed privacy framework can ensure relatively high intra-class privacy and utility, i.e. \( \Pi_2 > 0.63, Q > 0.70 \), at a privilege level range of \( 0.17 < l < 0.25 \). This is arguably acceptable as most non-critical MR functionalities may not necessarily require a unique object but only requires that the type or super-class of object be known. Moreover, since the unique identity (i.e. intra-class label) of the object is not exposed, the associated personally identifiable information with the unique object identity remains unexposed. Following the attacker descriptions from §5.2, as long as the user maintains a relatively low privilege level of \( l < 0.23 \), even with a high-privileged attacker like \( s_l \), we can expect that the intra-class privacy of the regenerations are kept relatively high \( \Pi_2 > 0.67 \) with a high upper bound for both utilities \( Q_1, Q_2 < 0.83 \).

**Temporal relationships of attacker success.** So far, we have primarily quantified the object reidentification privacy in terms of an attacker formulating a hypothesis to match a query point cloud object \( S^* \) with one or set of reference point cloud objects. Quantification was done based on the expected error in this hypothesis. On the other hand, another type of privacy threat could be posed in a situation where the attacker’s hypothesis on the query doesn’t correctly match with the corresponding original point cloud, but consistently matches with the same mismatching class over a large number of \( S^* \) queries from the same object. Here, although the object is not correctly reidentified, the attacker can generate a unique reference with respect to its previous queries and locate that object in a user’s surrounding. A successful privacy-based regeneration framework should be able to regenerate the privacy preserved version of the same object with enough variation between each iteration such that the possible misclassifications at the attacker’s end would not be consistent. We hope to do this analysis and evaluate the performance of our proposed framework in this context in future work.

**Point cloud segmentation-then-transformation.** Given a captured spatial mapping represented as a point cloud, relevant features in the point cloud needs to be detected and separated in order to process it further. For the proposed privacy framework, it is also necessary for the point cloud to be segmented and, then, separately transform the different objects to improve their privacy. In this work, we have directly explored and investigated the viability of the framework over already separated object point clouds. We intend to incorporate the actual segmentation task within a future iteration of this framework and apply the privacy transformations to these segmented point clouds.

### 8 CONCLUSION

Mixed reality and 3D processing technologies are steadily developing in parallel: advances in MR aim to improve the environmental understanding of the various viable devices that can be used for MR, while advances in 3D, particularly in machine learning, enables better processing of 3D data (i.e. 3D segmentation, classification, and identification). The interplay of the two technologies allows the rapid development and delivery of better and immersive MR experiences. However, at the same time, there is a rapid increase in the level of privacy risks that MR users are exposed to. These 3D processing algorithms along their with ever improving accuracy and discriminative power can be utilized by adversaries to reveal personally identifiable information from the 3D data that is captured and collected by MR devices and platforms.

In this work, we leverage these state-of-the-art 3D algorithms (1) to design a privacy framework for regenerating 3D point cloud data, and (2) to evaluate the viability of this framework using the latest 3D neural network-based classifiers. With the proposed privacy framework, varying the privilege level allows us to variably transform and, hence, effectively control the privacy exposure of the shared 3D data to potentially malicious third party applications. That is, as we increase the allowed privilege level \( l \), the privacy \( \Pi \) decreases, while the utility \( Q \) increases. Nonetheless, the proposed privacy framework is still able to ensure a relatively high intra-class privacy and utility if the privilege level is kept within \( 0.17 < l < 0.25 \). Therefore, despite not being able to always ensure the super-class privacy of a 3D object, the proposed privacy-aware 3D regeneration can prevent unique intra-class reidentification of 3D objects while maintaining high utility for common MR functionalities.
A APPENDIX

A.1 Privacy metric results for the top-1 success of Attacker $J_1$

For the completion of Fig. 7, here in Fig. 14 we show the privacy metric results of Attacker $J_1$ top-1 case for the remaining 6 classes.

Figure 14: Super-class $\Pi_1$ and intra-class $\Pi_2$ privacy measures for the top-1 success of attacker 1 for some 3D objects: (a) bathtub, (b) bookshelf, (c) cabinet, (d) lamp, (e) monitor, and (f) sofa.

A.2 Non-aggregated utility metrics for all super-classes

Fig. 15 shows all utility metrics including the Chamfer distance for every super-class.

Figure 15: Utility Plots

A.3 Privacy metric results for varying sizes of hypothesis sets for all the attackers

In §6.3 (as shown in Fig. 10), we discussed the aggregated variations in privacy metric as we vary the size of hypothesis sets $\rho$ for Attacker $J_1$. In Fig. 16, we display similarly aggregated $\Pi$ results for Attackers $J_2$, $J_3$, and $J_4$. 
Figure 17: Sample 3D regenerations from each super-class at different privilege levels.

A.4 Example plots of 3D regenerations
Fig. 17 shows example 3D point clouds for all super-classes regenerated at different privilege levels.

A.5 Example 3D plots for the RANSAC sitting plane for $Q_2$ utility
Fig. 18 shows example 3D point clouds for the RANSAC-generated sitting plane at different privilege levels.
User configurable 3D object regeneration for spatial privacy

Figure 18: RANSAC sitting plane ($\hat{\rho}$) and regenerated point cloud ($\hat{S}$) for Chair and Table classes at different privilege levels.