Learning to Censor by Noisy Sampling

Ayush Chopra[1], Abhinav Java[2], Abhishek Singh[1], Vivek Sharma[1], and Ramesh Raskar[1]

[1] Massachusetts Institute of Technology, Cambridge, MA
[2] Delhi Technological University, Delhi, India

Abstract. Point clouds are an increasingly ubiquitous input modality and the raw signal can be efficiently processed with recent progress in deep learning. This signal may, often inadvertently, capture sensitive information that can leak semantic and geometric properties of the scene which the data owner does not want to share. The goal of this work is to protect sensitive information when learning from point clouds; by censoring the sensitive information before the point cloud is released for downstream tasks. Specifically, we focus on preserving utility for perception tasks while mitigating attribute leakage attacks. The key motivating insight is to leverage the localized saliency of perception tasks on point clouds to provide good privacy-utility trade-offs. We realise this through a mechanism called *Censoring by Noisy Sampling* (*CBNS*), which is composed of two modules: i) Invariant Sampler: a differentiable point-cloud sampler which learns to remove points invariant to utility and ii) Noisy Distorter: which learns to distort sampled points to decouple the sensitive information from utility, and mitigate privacy leakage. We validate the effectiveness of CBNS through extensive comparisons with state-of-the-art baselines and sensitivity analyses of key design choices. Results show that CBNS achieves superior privacy-utility trade-offs on multiple datasets.

1 Introduction

Proliferation of 3D acquisition systems such as LiDARs, ToF cameras, structured-light scanners has made it possible to sense and capture the real-world with high fidelity. Point clouds are emerging as the preferred mode to store the outputs of these 3D sensors given that they are lightweight in memory and simple in form. Recent advances in deep learning have allowed to directly process the raw sensor output; which has enabled use of point clouds for diverse perception tasks across classification [1–6], semantic segmentation [7–10], object detection [11–14], and registration [15, 16]. This is facilitating algorithms for critical applications across autonomous navigation, precision surgery and secure authentication.

The deployment of downstream algorithms in these critical domains implies that the sensor often captures sensitive information, which the user would like to keep private. This is then inadvertently encoded in representations learned from the signal [17], leaking several semantic and geometric properties of the scene. Consider for instance, the robotic vacuum cleaners which use LiDAR sensors

to efficiently navigate inside the house. The captured signal is also sufficient to localize and map the entire house (via SLAM) as well as track and surveil individuals (via object detection). Similarly, this is also valid for the popular *face-id* experience in recent smartphones which use structured light to capture point clouds of the owner(s) face and use it for authentication, locally on-device. It is well understood that a lot of semantic information (age, gender, expression etc.) can be perceived from the point cloud - which the user may not be willing to share. With the emergence of strict regulations on data capture and sharing such as HIPAA [18], CCPA [3], capturing such sensitive information can create legal liabilities. The goal of this paper is to alleviate such privacy concerns, while preserving utility, by transforming the point cloud to censor sensitive information *before* it is released for downstream utility tasks.

In practice, the design of such transformation functions depends upon the definition of the utility task and privacy attack. Most prior work has focused on preserving the utility of geometric tasks (image-based localization, SLAM, SfM, etc.) while protecting against input reconstruction attacks [19]. For these setups, the dominant idea is to transform the point cloud into 3D line cloud [20] which obfuscates the semantic structure of the scene while preserving utility for camera localization [21], SLAM [22], SfM [23] etc. In contrast, we focus on providing utility for perception tasks (classification, detection, segmentation etc.) while mitigating sensitive attribute leakage [24]. We posit that projecting to line clouds is an infeasible transformation for perception tasks because: i) line clouds disintegrates the semantic structure of the scene required for perception which worsens the utility. This is visualized in [20] and validated by our analysis in section 6; and ii) line clouds are now also vulnerable to inversion attacks, as recently shown in [25], which worsens the privacy. We propose *Censoring by Noisy Sampling (CBNS)* as an alternate transformation for censoring point clouds, which provides improved privacy-utility trade-offs.

The motivating insight for *CBNS* is that performance on perception tasks (utility) only depends upon only a small subset of points (critical points) such that removing (or *sampling*) other non-critical points does not change prediction. Leveraging this for censoring point clouds presents two challenges: *First*, conventional point cloud sampling methods are designed to improve compute efficiency while retaining maximal information about a specific task. Hence, we need to design methods that can jointly sample critical points for the utility task and remove information *invariant* to utility. *Second*, this invariant sampling is necessary but not sufficient, as critical points for task and sensitive attributes can overlap; as we observe through quantitative analysis in section 3.1. We develop *CBNS* to overcome these challenges - i) by introducing an invariant sampler that balances privacy-utility trade-off in its sampling via an adversarial contrastive objective (ℓ_{aco}); ii) by designing a noisy distortion network that adds sample-specific noise to minimize the overlap between task and sensitive information in

[3] `https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375`

an utility conducive manner. We demonstrate the effectiveness of our solution in section 4.

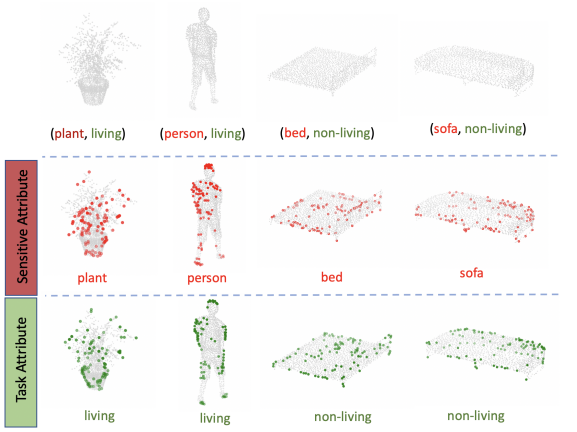
Contributions: Our CBNS is an end-to-end learning framework for protecting sensitive information in perception tasks by dynamically censoring point clouds. CBNS is composed of: i) an invariant sampler that learns to sample non-sensitive points in a task-oriented manner by balancing privacy-utility, ii) a noisy distorter that learns to randomize sampled points for utility conducive removal of sensitive information. We demonstrate the effectiveness of our framework through extensive comparisons against strong baselines and analyses of key design choices. Results show that CBNS significantly improves privacy-utility trade-offs on multiple datasets.

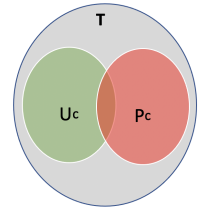
2 Problem Formulation

This section formalises the notation for our task, the threat and attack models, and our privacy definition.

Notation: Consider a data owner O with a point cloud dataset $D_O = (P, Y)$ of N datapoints and (p, y) denotes a paired sample, for $p \in P$ and $y \in Y$. Specifically, $p \in R^{m \times d}$ is a *point cloud* defined as an unordered set of m elements with d features; and y is a *label set* of k attributes describing p. For instance, p can be a 3D point cloud representing a human face ($p \in R^{m \times 3}$) with the *set* y containing categorical attributes that indicate the {*age, gender, expression*} ($k = 3$) of the individual. For every pair $(p, y) \in D_O$, certain attributes in the label set y represent sensitive information which the data owner (O) wants to keep private (y_s) but is comfortable sharing the non-sensitive (or task) information (y_t), such that ($y = y_s \cup y_t$). The risk of leaking this sensitive information prevents the data owner from sharing D_O with untrusted parties; especially with recent progress in deep learning where attackers can efficiently learn functions (F) that can directly map the raw point cloud p to any attribute $a \in y$, where $a = F(p)$ [14, 6]. Trivially omitting y_s from y to share the dataset of paired samples $\{(p, y_t)\}$ is not enough since the sensitive information is encoded in p and can be inferred by attackers (*e.g.* using pre-trained models or auxilliary datasets). Hence, to facilitate data sharing with untrusted parties, it is essential to *censor* the sensitive information in p (that leaks y_s) *before* the dataset can be released. Our goal is to learn such a transformation function ($T(\theta_T; \cdot)$) that censors each sample in D_O by generating $\hat{p} = T(p)$. This allows to release $(\hat{p}, y_t)$ instead of (p, y_t). Henceforth, we denote this *censored dataset* as $(\hat{P}, \hat{Y})$. The key challenge for T is to preserve utility of the task information (y_t) while protecting privacy of sensitive information (y_s). In practice, the design of T depends upon definition of the utility task and privacy attack. We focus on providing utility for perception tasks while mitigating attribute leakage attacks [24].

Threat Model: We assume that the attacker gains access to a subset of censored point clouds $(\hat{P}, Y)$ intending to infer sensitive attributes (y_s). This is practical since data owners typically share data with external entities for storage and also for monetary incentives. Further, the threat is also valid if the attacker gains





Critical Points	mIoU (task, sensitive)	
	ModelNet	FaceScape
50	0.171	0.117
64	0.215	0.128
100	0.315	0.223
500	0.663	0.572

Fig. 1: **Premise Validation**. (*Left*) - perception on point clouds depends upon few critical points. (*Right - bottom*) table shows overlap of critical points for sensitive (P_c) and task (U_c) attributes; (*Right - top*) goal of a censoring mechanism is to remove $T - U_c$ and reduce $P_c \cap U_c$. We bridge these ideas in section 3.1 and introduce CBNS in section 3.2 to accomplish both goals.

access to the learned censoring function T which can be used to simulate a dataset that mimics the censored point cloud distribution. This is practical if the attacker is one of the data owners that has access to T. We note that unlike differential privacy [26] that protects identifiability, our threat model protects sensitive attribute leakage [24].

Attack Model: We model an attacker that uses the released dataset to train state-of-the-art DNN models that can directly predict the sensitive attribute from the point cloud. This attacker may use arbitrary models which are not accessible during training, and hence we mimic a proxy attacker for learning the censoring transformation. We represent the proxy attacker by a state-of-the-art DNN parameterized as $f_A(\theta_A; \cdot)$ and trained on censored point clouds $\hat{P}$.

Privacy: Following the setup described by Hamm *et al.* [27], we define privacy as the expected loss over the estimation of sensitive information by the attacker. This privacy loss L_{priv}, given ℓ_p norm, for an attacker can be stated as:

$$L_{priv}(\theta_T, \theta_A) \triangleq E[\ell_p(f_A(T(p; \theta_T); \theta_A), y_s)]$$

Under this definition, releasing sensitive information while preserving privacy manifests as a *min-max* optimization between the data owner and the attacker. However, for training the model parameters, we use a proxy adversary from which gradients can be propagated. We refer to the attack performed by this proxy attacker as an *online attack* and note that this allows mimicking worst-case setups where the attacker can also dynamically adapt using protected data and sensitive label information [28]. We note that our definition of privacy sig-

nificantly differs from differential privacy [26] since we aim to protect sensitive attributes instead of the identity of the data owner.

3 Methodology

In this section, we introduce *Censoring by Noisy Sampling (CBNS)* - a mechanism to censor point clouds for enabling utility of perception tasks while protecting leakage attack on sensitive attributes. We begin by discussing our key motivating insight and then delineate the proposed *CBNS* mechanism.

3.1 Premise Validation

State-of-the-art DNN models such as PointNet [1], PointNet++ [2], DGCNN [3] have successfully handled the irregularity of the raw point cloud and achieved remarkable progress on perception tasks such as classification, segmentation etc. Extensive empirical analysis of these networks shows that classification performance depends upon only a small subset of points (*critical points*) such that removing other non-critical points does not change prediction. Figure 1 visualizes the critical points for perceiving the category (*plant, person, bed, sofa*) and super-type (*living, non-living*) of a few ModelNet dataset samples [29] by training PointNet. The observed *localized* (i.e. depends on critical points) and *task-oriented* (different across category and super-type) saliency is a key motivating insight for censoring point clouds for privacy-utility release.

Assume a privacy-utility scenario where the super-type is task (utility) and the category is the sensitive attribute (privacy). In principle, we achieve good utility (predicting super-type) by only keeping the necessary critical points. Since critical points are visualized via post-training analysis, in practice, this presents two challenges for data release: *First,* conventional point cloud sampling methods are designed to improve compute efficiency while retaining maximal information for a specific task. Hence, we need to design methods that can jointly sample critical points for the utility task and remove information *invariant* to utility. *Second*, this invariant sampling is necessary but not sufficient, as critical points for task and sensitive attributes can overlap. For instance, the top-100 critical points for super-type and category in ModelNet have mIoU of *31%* (table in figure 1). Hence we also need to distort the sampled points to decouple the sensitive and task attributes. The venn-diagram in figure 1 helps visualize this constraint. Intuitively, we want to learn a censoring transformation that can concurrently remove $T - U_c$ and reduce $P_c \cap U_c$. With this motivation, next we describe our proposed mechanism to censor point clouds.

3.2 Censoring by Noisy Sampling

The task of censoring to mitigate information leakage involves three key entities: i) Data Owner (O), ii) User (U) and iii) Attacker (A). O censors each sample in the dataset to protect sensitive information and releases it for U, an untrusted

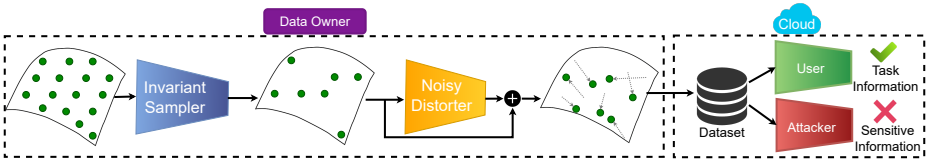


Fig. 2: **Censoring by Noisy Sampling** through a a three-player game: i) Data Owner (O), ii) User (U) and iii) Attacker (A). O censors every sample in the dataset and shares it with U to train a model on the task information. A intercepts the released dataset and attempts to leak the sensitive information. We design *CBNS*, composed of two modules: a) *Invariant Sampler* and b) *Noise Distorter*, to help O enable U's task and avert A's attack. The design of the mechanism is delineated in section 3.2.

but honest entity, to learn a model on the non-sensitive information. A intercepts the released dataset and queries it to leak the sensitive information. We design *Censoring by Noisy Sampling* (**CBNS**) to help O facilitate the task of U and prohibit the task of A. This three-player game [30] is summarized in figure 2 and described below:

a) Owner owns the point cloud dataset (P, Y) which is to be released. This entity censors the sensitive information in each sample (p, y) for $p \in P$ and $y \in Y$. CBNS is composed of two parametric modules, applied sequentially: i) *Invariant Sampler* ($f_S(\theta_S; \cdot)$), ii) *Noisy Distorter* ($f_D(\theta_D; \cdot)$).

First, $p \in R^{m \times d}$ is passed through $f_S(\theta_S; \cdot)$, differentiable DNN sampler built upon [31], which selects a subset of r points relevant for encoding task information to generate an intermediate point cloud $p_s \in R^{r \times d}$, where $r << m$. In contrast to conventional sampling methods which are aimed at improving compute efficiency while preserving all information, f_S is a lossy sampler designed to remove points *invariant* to utility. The extent and quality of censoring depends upon design of f_S, which we analyse in section 6. Releasing this p_s may still leak sensitive attribute through points which overlap with utility (mIoU table in fig 1). *Next*, p_s is passed through $f_D(\theta_D; \cdot)$ which generates task-oriented noise to distort p_s. This is done to decouple overlapping sensitive (privacy) and task (utility) information and executed in the following steps: i) $\mu, \sigma = f_D(p_s; \theta_D)$, ii) $z_s \sim \mathcal{N}(\mu, \sigma^2)$, iii) $\hat{p} = p_s + z_s$ where the sampled noise $z_s \in R^{r \times d}$. All the censored point cloud samples $(\hat{p}, \hat{y})$ are aggregated into the dataset $(\hat{P}, \hat{Y})$ and then released for use by untrusted parties.

b) User is an untrusted but honest entity that receives the released dataset $(\hat{P}, \hat{Y})$ and uses it to infer the non-sensitive task attributes. The *User* trains a state-of-the-art DNN model that can learn to directly map the raw point cloud signal to the task attribute (y_t). For training *CBNS*, we mimic the real user using a proxy user which is parameterized with $f_U(\theta_U; \cdot)$ that consumes $\hat{p} \in \hat{P}$ to predict $y_t \in \hat{Y}$.

c) Attacker is an untrusted semi-honest entity that acquires access to $(\hat{P}, Y)$ with the intention to leak sensitive information about the data owner. The attacker is parameterized with $(f_A(\theta_A; \cdot))$ which is not accessible during design of *CBNS*. Hence, for training, we use a proxy attacker that consumes $\hat{p} \in \hat{P}$ to predict the sensitive attribute (y_s). We note that f_A is a proxy attacker used for training *CBNS*, while a *distinct offline attacker* (f), not used for training, is employed for evaluation tasks.

Training: The utility loss is approximated based on the performance of the proxy user which depends upon parameters θ_T $(\theta_T = \theta_S \cup \theta_D)$ and θ_U that are learned during training. The objective function is given by:

$$L_{util}(\theta_S, \theta_D, \theta_U) \triangleq E[\ell_u(f_U(f_D(f_S(p; \theta_S)\theta_D); \theta_U), y_t)] \tag{1}$$

where, θ_S, θ_D are parameters of CBNS and θ_U are parameters of the the proxy user network; and ℓ_u is the cross entropy loss (ℓ_{cce}).

The privacy loss is approximated based on the performance of the proxy attacker which depends upon parameters θ_T, θ_A learned during training. The objective function is given by:

$$L_{priv}(\theta_S, \theta_D, \theta_A) \triangleq E[\ell_a(f_A(f_D(f_S(p; \theta_S)\theta_D); \theta_A), y_s)] \tag{2}$$

where, θ_S, θ_D are parameters of CBNS and θ_A are parameters of the proxy attacker; and ℓ_a denotes the attacker loss. For training CBNS, we define ℓ_a with the following objective function:

$$\ell_a = \alpha * (\ell_{cce}(f_A(f_D(p_s)), y_s)) + (1 - \alpha) * \ell_{aco}(f_D(p_s), y_s, y_t) \tag{3}$$

where ℓ_{cce} is categorical cross-entropy and ℓ_{aco} is an adversarial contrastive loss, inspired from [32]; and α is a scalar hyperparameter.

Adversarial Contrastive Loss (ℓ_{aco}): Our analysis in section 6 shows that using ℓ_{aco} significantly improves privacy-utility trade-offs. Consider, for instance, age (y_s) to be the sensitive attribute. In the conventional contrastive loss, we encourage to pull positive samples (same age) closer within the local neighborhood and negative samples (different age) apart. In contrast, ℓ_{aco} pulls negative samples closer (different age) and positive samples (same age) apart. The goal here is to map all different ages within a very small neighborhood of each other, to deter the attacker from learning discriminative representations of age. Intuitively, this guides *CBNS* to transform the released point cloud to introduce ambiguity in representations used by an attacker to correctly discriminate between ages, resulting in better privacy. In other words, the ℓ_{aco} forces to map the different ages to a single point in the embedding space.

The proxy attacker and proxy user have access to supervised data and attempt to minimize their losses L_{util} and L_{priv} respectively. CBNS is trained to minimize L_{util} and maximize L_{priv}, simulating an implicit min-max optimization for these two components. Furthermore, CBNS also minimizes a soft-projection loss [31] (L_{sample}) to improve stability of f_S and ensure that the sampler is

constrained to *select* points from the input set (instead of interpolating). This overall objective can be summarized as:

$$\min_{\theta_S,\theta_D}\left[\max_{\theta_A} L_{priv}(\theta_S,\theta_D,\theta_A)+\lambda \min_{\theta_S,\theta_D,\theta_U} L_{util}(\theta_S,\theta_D,\theta_U)+\min_{\theta_S} L_{sample}(\theta_S)\right] \tag{4}$$

Here, λ is a chosen hyperparameter to help regulate the trade-off between privacy and utility.

Inference: A data owner with access to a point cloud dataset (P, Y) can use CBNS to generate the censored dataset $(\hat{P}, \hat{Y})$ and release it for use by untrusted parties. This released dataset can be used for either: i) training new models or ii) running inference using pre-trained models. This is possible only because the output space of the censoring mechanism is same as the input space. In other words, censoring a point cloud using CBNS also generates a point cloud. In contrast: i) most work for censoring images requires releasing neural activations which cannot be processed by arbitrary designed networks [28, 33, 32] and ii) prior work for censoring point clouds releases line clouds [21, 20] which, while useful for geometric tasks, are incompatible for off-the-shelf perception networks.

4 Experiments

In this section, we specify the datasets and baselines used, define the evaluation protocols and summarize implementation details for the results presented in this work. Details about the code are included in the appendix.

Datasets: a) FaceScape [34] consists of 16,940 textured 3D faces, captured from 938 subjects each with multiple categorical labels for age (100), gender (2) and expression (20). For our experiments, we sample 1024 3D points from the surface of each face mesh using Pytorch 3D [35]. To simulate privacy-utility analysis, we use the expression as the task attribute (utility) and gender as the sensitive attribute (privacy). We choose this configuration because the default critical points for the two attributes overlap (need for noisy distorter) and are also distributed across the point cloud (need for invariant sampling), but human performance motivates that they are can be inferred independently. This provides a good benchmark for testing the efficacy of CBNS. **b) ModelNet** [29] consists of 12,311 CAD-generated meshes across 40 categories (object types) of which 9,843 training and 2,468 testing data points. For our experiments, we uniformly sample 2048 3D points from the mesh surface and then project them onto a unit sphere. Since each input sample only has one attribute (object type), we adapt the strategy used by [30] to simulate our privacy-utility analysis. Specifically, to identify an additional attribute, we divide the 40 classes into two super-types: living and non-living. We anticipate living objects to have visually discriminative features instead of geometric shapes of non-living objects. For example, the task of classifying an object as living (*person, plant*) or non-living (*sofa, bed*) should not reveal any information about its underlying identity (*person, plant, sofa,*

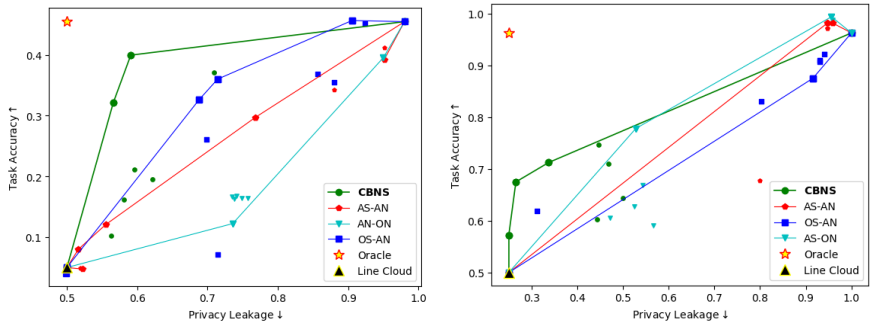
bed). We use super-types as task attribute and object type as sensitive attribute.

Baselines: Prior work in censoring point clouds has largely focused on geometric tasks (image-based localization, SLAM, SfM, etc) via line clouds [20, 21, 23, 22]. However, our analysis (section 6) shows that line clouds are a weak baseline for perception tasks (classification, detection, etc). To ensure rigorous analysis, we define multiple baselines inspired by work in 2D vision that has focused on censoring images (and their activations) for perception tasks while mitigating attribute leakage. The baselines differ in the design of sampling and noisy distorter modules - which may be task-oriented (learned using data) or task-agnostic (deterministic). Our mechanism *CBNS* is equivalent to the *Oriented Sampling - Oriented Noise (OS-ON)* configuration. The baselines are summarized below, with more details in the appendix:

- **Agnostic Sampling - Agnostic Noise (AS-AN):** Uses farthest point sampling (FPS) with fixed gaussian distribution for noise. This is inspired from [36] which formalises differential privacy for images through Gaussian noise, without any learning. While [36] adds noise to images, we add it to a sampled point cloud obtained via FPS.
- **Agnostic Sampling - Oriented Noise (AS-ON):** Uses FPS and learns parameters of the gaussian distribution for noise (as in CBNS) using *maximum likelihood* attacker training. This is inspired from [33] which learn a noise distribution to obfuscate image activations. While [33] adds noise to image activations, we add it to a sampled point cloud obtained via FPS.
- **Oriented Sampling - Agnostic Noise (OS-AN):** Uses differentiable point cloud sampling (as in CBNS) with fixed gaussian distribution for noise. This is inspired from [28] which does channel pruning of image activations to remove sensitive information. While [28] trains a DNN to prune neural activations, we train a DNN to sample point clouds [31].

Evaluation Protocol: We evaluate different techniques by comparing the *privacy-utility trade-off*. For each technique, utility is a measure of *User*'s performance by training on the *censored* dataset and privacy is of the *Attacker*'s performance (as described in section 2). Specifically, we quantify information leakage from the dataset by comparing the performance of an attacker to correctly infer sensitive information from the censored dataset. For this analysis, we simulate a *worst case* attacker that dynamically adapts to the privatization scheme. This adaptation is modeled using a pretrained attacker model that is fine-tuned on the *censored* dataset and then evaluated on a censored test set. Inspired by [30], we quantify privacy-utility trade-offs curves by different techniques using area under the pareto-optimal curve denoted as the normalized hypervolume (**NHV**) [37]. *Higher NHV value indicates a better privacy-utility trade-off.*

Implementation Details: Unless stated otherwise, we use 3D-point clouds ($d = 3$) with the Invariant Sampler producing 64 points ($m = 64$) and with PointNet [1] as the backbone architecture for both f_U and f_A. For FaceScape, $n = 1024$ and the dataset split is 80% training and 20% testing examples. For ModelNet, $n = 2048$ and we restrict experimentation to a smaller set of 4 classes:



(a) FaceScape [34] (b) ModelNet [29]

Fig. 3: **Privacy-Utility trade-off** comparison for different techniques by interpolating their best performing points. The oracle point refers to the best possible censoring mechanism. We note that the line cloud [20, 22] techniques do not yield any trade-off due to its incompatibility with perception tasks.

2 living and 2 non-living to ensure ease of analysis and avoid data imbalance issues (ModelNet has 2 living and 38 non-living objects). All experiments are implemented using Pytorch and conducted on 2 TITAN-X GPUs. We will release both our code and dataset splits for reproducibility.

5 Results

We report performance comparison with baselines on ModelNet and FaceScape, in Table 1. For completeness, apart from the baselines defined in section 4, we also benchmark with two extreme scenarios: i) **No-privacy**: default case where the data is released without any censoring and, ii) **Oracle**: best possible case with some ideal censoring mechanism which *does not exist*. Results show that **CBNS** significantly outperforms all baselines, on both datasets; as evident from higher NHV. *First,* **CBNS** consistently provides the best privacy leakage - often very close to random chance. Specifically, privacy leakage with CBNS is 0.5890 for 2-way classification in FaceScape and 0.2657 for 4-way classification in ModelNet. *Second,* the peak privacy-utility trade-off for **CBNS** is closest to the oracle, on both datasets. Specifically, when CBNS is used for data release on FaceScape, the *User* achieves a utility of *0.4013* (88% of the oracle) while the *Attacker* performance is close to random chance (0.5890). This corresponds to *13% less privacy leakage while also providing 4% more utility* than the closest baseline (OS-AN). However, a higher fall in utility is observed on ModelNet, which can be attributed to the fact that the task and sensitive attributes are more strongly coupled, than in FaceScape. *Finally,* these observations are corroborated visually by the privacy-utility trade-off curves in Figure 3 where CBNS has the highest area-under-curve (correlated with the hyper-volume).

Method	FaceScape			ModelNet		
	Privacy (↓)	Utility (↑)	NHV (↑)	Privacy (↓)	Utility (↑)	NHV (↑)
No-privacy	0.9515	0.4549	-	0.95	0.9625	-
Line Cloud [20, 22]	0.5000	0.0500	-	0.2500	0.5000	-
AS-AN [36]	0.9511	0.4122	0.1524	0.9469	0.9612	0.6113
AS-ON [33]	0.9391	0.3875	0.1250	0.5281	0.7781	0.6088
OS-AN [28]	0.7143	0.3602	0.1439	0.3125	0.6187	0.6356
CBNS (Ours)	**0.5890**	**0.4013**	**0.1885**	**0.2657**	**0.6750**	**0.6492**
Oracle	0.5000	0.4549	-	0.2500	0.9625	-

Table 1: **Comparison for sensitive attribute leakage**. We compare our approach on sensitive attribute leakage with the existing works and baseline. CBNS outperforms Line Cloud [20, 22], AS-AN [36], AS-ON [33], OS-AN [28] and achieves the best privacy-utility trade-off on FaceScape and ModelNet datasets. For the FaceScape, sensitive attribute is gender and task attribute is expression; and for the ModelNet, sensitive attribute is underlying object type and task attribute is super-types (living or non-living).

Technique	Privacy (↓)	Utility (↑)	NHV (↑)
AS-ON [33] (ℓ_{cce})	0.9297	0.3954	0.1322
OS-AN [28] (ℓ_{cce})	0.6942	0.3602	0.1439
CBNS (ℓ_{me})	0.6839	0.413	0.1596
CBNS (ℓ_{cce})	0.5787	0.3638	0.1615
CBNS (ℓ_a)	**0.5707**	**0.3997**	**0.1885**

Table 2: **Design of Invariant Sampler**. Privacy-utility trade-off is influenced by whether transformation is learned, and how it is learned. For FaceScape, sensitive attribute is gender and task attribute is expression. l_{cce}, l_{me} and l_a denotes cross entropy loss, max-entropy loss and CBNS loss respectively.

6 Discussion

We analyse the impact of key design choices for censoring by noisy sampling. Specifically, we study the characterization of both *CBNS* modules: i) Invariant Sampler and ii) Noisy Distorter; and impact of the perception network. For completeness, we also analyse the viability of line clouds for perception tasks. For ease of exposition, we restrict the scope of this analysis to FaceScape dataset.

– **Design of Invariant Sampler:** We study the role of two design choices: i) learning a task-oriented sampler and ii) the attacker loss that is used to optimize the parameters of the learned sampler. Results are presented in Table 2. Please note that we follow the baselines from section 4 and explicitly mention (L_{priv}) the attacker objective in parenthesis. Specifically, l_{cce} is the cross-entropy loss, l_a is our proposed loss (equation 3) and l_{me} is a max-entropy loss used in [30]. We observe the following: *First*, using a learned task-oriented sampler reduces

Technique	Privacy (↓)	Utility (↑)	NHV (↑)
OS (ℓ_{cce})	0.5787	0.3638	0.1615
OS-AN [28] (ℓ_{cce})	0.6942	0.3602	0.1439
CBNS (*shared*, ℓ_{cce})	0.9051	0.4532	0.1586
CBNS (*shared*, ℓ_a)	0.8600	0.4217	0.1625
CBNS (*pointwise*, ℓ_{cce})	0.5689	0.4013	0.1530
CBNS (*pointwise*, ℓ_a)	**0.5707**	**0.3997**	**0.1885**

Table 3: **Design of Noisy Distorter**. Privacy-utility trade-off is influenced by the learned noise and the granularity of the noise parameters (*shared v.s pointwise*). For FaceScape, sensitive attribute is gender and task attribute is expression. l_{cce} and l_a denotes cross entropy loss and CBNS loss respectively.

privacy-leakage by 35% without any loss to utility (row 1 vs 4 and 5) in contrast to a task-agnostic sampler. *Second,* using the proposed adversarial contrastive loss in L_{priv} improves privacy-utility trade-off by increasing utility by 3% without any additional privacy leakage (row 4 vs 5). *Third,* using l_{me} improves utility but with a significant increase in privacy-leakage; as evident from lower NHV. l_{me} is successful for images [30] but fails to generalize to point clouds, which can be attributed to the irregularity in the data structure.

– **Design of Noisy Distorter:** We study the role of three design choices: i) learning task-oriented noise, ii) the attacker loss used to optimize parameters of the learned noise (l_a or l_{cce}) and iii) the granularity of the noise parameters (*shared v.s pointwise*). *Shared* implies that each point is distorted using noise from the same learned distribution (i.e. $z_s \in R^1$) and *Pointwise* implies that each point is distorted from a unique independently learned distributions (i.e. $z_s \in R^r$). Results are presented in Table 3. We follow the baselines from section 4 and define *OS* (row 1) as additional baseline which only uses sampling (*without* noisy deformation). We observe the following: *First,* using noise task-agnostic (row 1 vs 2), or ii) shared task-oriented (row 1 vs 3, 4) does not provide benefit; and are infact worse than no noise baseline (*SO*). *Second, pointwise* noise distributions *significantly* improves performance. (row 3 vs 5; 4 vs 6). This increase in NHV as well as peak privacy-utility trade-offs can be attributed to improved flexibility for adapting to characteristics of sensitive, task attributes and their relationship.. *Third,* the objective function used for learning noise is also important where using l_{aco} in l_a improves privacy-utility trade-off (row 3 vs 4; row 5 vs 6).

– **Impact of Perception Network:** In our threat model, the *Owner* releases dataset for post-hoc access by the *User* and *Attacker*. Hence, the censoring mechanism should be independent of the type of perception networks used by these entities for downstream tasks. We analyse this sensitivity by comparing two different proxy attacker and user networks. Specifically, we use DGCNN [3] which is a recent state-of-the-art network with higher capacity and distinct saliency properties than PointNet [38]. Results are presented in Table 4. We observe the following: *First,* increasing capacity of proxy networks further improves learning of CBNS as evident from better privacy-utility trade-offs (row 1 vs 2; row 3 vs

Technique	Backbone	Privacy (↓)	Utility (↑)	NHV (↑)
OS-AN [28]	PointNet	0.6942	0.3602	0.1439
	DGCNN	0.7056	0.4718	0.2136
CBNS (Ours)	PointNet	0.5707	0.3997	0.1885
	DGCNN	**0.4848**	**0.4100**	**0.2361**
Line Cloud [20, 22]	PointNet	0.5000	0.0500	-
	DGCNN	0.5000	0.0500	-

Table 4: **Impact of Perception Network** and **Incompatibility of Line Clouds**. For FaceScape, sensitive attribute is gender and task attribute is expression. CBNS is invariant to the type of attacker network. Using stronger perception network (DGCNN) further improves performance over PointNet and helps achieve near optimal trade-off with our proposed CBNS. Resampling line clouds provides poor (random chance) privacy-utility trade-off.

4). For instance, when CBNS is trained with DGCNN (as against PointNet), the censored dataset provides a better utility of 0.4100 (vs 0.3997) while also significantly reducing privacy leakage to 0.4848 (vs 0.5707). *Second,* importantly, we see that noisy sampling is independent of the downstream network and can generalize to multiple perception networks. Specifically, this is very encouraging since CBNS can concurrently mitigate stronger *attackers* from leaking information by improving the utility of *users* with the stronger perception backbones.

– **Incompatibility of Line Clouds:** We posit that line clouds are an incompatible baseline for perception tasks because i) they destroy semantic structure of the input point cloud which is essential for perception (see visualizations in [20]), and ii) any off-the-shelf perception network: used by both *User* and *Attacker* cannot train on line clouds. To benchmark the performance of line clouds, we generate point clouds by re-sampling line clouds and evaluate performance on our perception queries. Results in Table 4 show that: i) re-sampling line clouds provides extremely poor utility (random chance), and ii) the privacy-utility trade-off cannot be tuned (hence no NHV). Specifically, we observe that resampled line cloud obtain privacy leakage of 0.5000 (for 2-way classification) and utility of 0.05 (on a 20-way classification). Finally, we acknowledge recent work has attacked line clouds to reconstruct point clouds [25] but note that this is equivalent to our No-Privacy baseline which can improve utility but requires mechanisms like noisy sampling to provide privacy.

7 Related Work

Private Imaging. A majority of the existing works in privately sharing data focus on identifiability and anonymization [39, 26, 40]. In contrast to this line of work, we focus on protecting sensitive attributes. Among the techniques that focus on protecting sensitive attributes [33, 28, 41, 32, 42], their tasks are typically limited to image datasets. More recently, privacy for 3D point clouds has

emerged with a focus on geometric queries protecting privacy by releasing line clouds [20, 23]. However, To the best of our knowledge, this is the first work in protecting sensitive information leakage for perception tasks in point clouds. Adjacent to research in privately sharing data, privately sharing ML model [43–47] has received interest recently. However, unlike protecting sensitive attributes, model sharing aims to protect the identifiability of training data.

Learning on Point Clouds. Recent advances in deep learning (DL) have allowed to learn directly on raw point clouds; enabling use in diverse perception tasks such as classification, semantic segmentation, object detection, registration etc. Various DL architectures have been proposed starting with PointNet [1], PointNet++ [2] and follow-up works in [48, 8, 49–53, 5, 4] that improve the performance over a given task by capturing task-oriented representations. Zheng *et al.* [38] observe that saliency of the point cloud networks is localized and network rely on a small subset of the signal for the task. This observation has led to extensive work in privacy and security [54–57] utilizing the localized saliency used to design adversarial attack (and defence) mechanisms on the trained models. We note that our setting significantly differs from adversarial attack work since we *protect the dataset* that can be used to train arbitrary models while adversarial methods focus on *attacking/protecting* the robustness of model predictions.

Sampling of Point Clouds. Processing point clouds can be computationally intensive making sampling a popular pre-processing step to alleviate this challenge. Classical methods such as random sampling and FPS [2, 4] are task-agnostic and deterministic algorithms for sampling point sets. However, not utilizing task knowledge when sampling hinders performance. Recent techniques [58, 31] introduce task-oriented mechanisms for sampling through differentiable approximations. The focus is to improve compute efficiency while preserving the entire signal in the sampled subset. In contrast, our goal is to censor sensitive information during the sampling process. We build upon prior work to introduce a task-oriented point-cloud sampler that censors sensitive information.

Noisy Sampling for Censoring. While not motivated for point clouds, similar intuition has been used for tabular datasets for private coresets [59, 60] combines subset (coreset) selection and differentially-private noise to achieve good privacy utility trade-off. Our work is different because: i) our queries involve neural networks so computing sensitivity for DP-noise is infeasible, ii) we only want to protect sensitive attribute. We empirically validate privacy-utility trade-off using benchmark metrics, as described in section 4 and present results in section 5.

8 Conclusion

This focus of this paper is to censor point clouds to provide utility for perception tasks while mitigating attribute leakage attacks. The key motivating insight is to leverage the localized saliency of perception tasks on point clouds to provide good privacy-utility trade-offs. We achieve this through our mechanism called censoring by noisy sampling (*CBNS*), which is composed of two modules: i) Invariant Sampling - a differentiable point-cloud sampler which learns to remove points invariant to utility and ii) Noise Distorter - which learns to distort sampled

points to decouple the sensitive information from utility, and mitigate privacy leakage. We validate the effectiveness of CBNS through extensive comparisons with state-of-the-art baselines and sensitivity analyses of key design choices. Results show that CBNS achieves superior privacy-utility trade-offs.

References

1. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation (2017)
2. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
3. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5) (2019) 1–12
4. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. Advances in neural information processing systems **31** (2018)
5. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 9621–9630
6. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 16259–16268
7. Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In: European Conference on Computer Vision, Springer (2020) 1–19
8. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 11108–11117
9. Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H.: Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 9601–9610
10. Chen, C., Qian, S., Fang, Q., Xu, C.: Hapgn: Hierarchical attentive pooling graph network for point cloud segmentation. IEEE Transactions on Multimedia **23** (2020) 2335–2346
11. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 918–927
12. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 9277–9286
13. Martin, S., Stefan, M., Karl, A., et al.: Complex-yolo: real-time 3d objectdetection on point clouds. In: Computer vision and pattern recognition. (2018)
14. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2019) 770–779

15. Sarode, V., Li, X., Goforth, H., Aoki, Y., Srivatsan, R.A., Lucey, S., Choset, H.: Pcrnet: Point cloud registration network using pointnet encoding. arXiv preprint arXiv:1908.07906 (2019)
16. Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: Robust & efficient point cloud registration using pointnet. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 7163–7172
17. Song, C., Shmatikov, V.: Overlearning reveals sensitive attributes. arXiv preprint arXiv:1905.11742 (2019)
18. Atchinson, B.K., Fox, D.M.: From the field: the politics of the health insurance portability and accountability act. Health affairs **16**(3) (1997) 146–150
19. Arora, S., Liang, Y., Ma, T.: Why are deep nets reversible: A simple theory, with implications for training. CoRR **abs/1511.05653** (2015)
20. Speciale, P., Kang, S.B., Pollefeys, M., Schönberger, J., Sinha, S.: Privacy preserving image-based localization. In: 2019 Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (June 2019)
21. Speciale, P., Schonberger, J.L., Sinha, S.N., Pollefeys, M.: Privacy preserving image queries for camera localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 1486–1496
22. Shibuya, M., Sumikura, S., Sakurada, K.: Privacy preserving visual slam. In: European Conference on Computer Vision, Springer (2020) 102–118
23. Geppert, M., Larsson, V., Speciale, P., Schönberger, J.L., Pollefeys, M.: Privacy preserving structure-from-motion. In: European Conference on Computer Vision, Springer (2020) 333–350
24. Jia, J., Gong, N.Z.: {AttriGuard}: A practical defense against attribute inference attacks via adversarial machine learning. In: 27th USENIX Security Symposium (USENIX Security 18). (2018) 513–529
25. Chelani, K., Kahl, F., Sattler, T.: How privacy-preserving are line clouds? recovering scene details from 3d lines (2021)
26. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference, Springer (2006) 265–284
27. Hamm, J.: Minimax filter: Learning to preserve privacy from inference attacks. Journal of Machine Learning Research **18**(129) (2017) 1–31
28. Singh, A., Chopra, A., Sharma, V., Garza, E., Zhang, E., Vepakomma, P., Raskar, R.: Disco: Dynamic and invariant sensitive channel obfuscation for deep neural networks (2021)
29. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 1912–1920
30. Roy, P.C., Boddeti, V.N.: Mitigating information leakage in image representations: A maximum entropy approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 2586–2594
31. Lang, I., Manor, A., Avidan, S.: Samplenet: Differentiable point cloud sampling (2020)
32. Osia, S.A., Shahin Shamsabadi, A., Sajadmanesh, S., Taheri, A., Katevas, K., Rabiee, H.R., Lane, N.D., Haddadi, H.: A hybrid deep learning architecture for privacy-preserving mobile analytics. IEEE Internet of Things Journal **7**(5) (May 2020) 4505–4518
33. Mireshghallah, F., Taram, M., Ramrakhyani, P., Jalali, A., Tullsen, D., Esmaeilzadeh, H.: Shredder: Learning noise distributions to protect inference pri-

vacy. In: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. (2020) 3–18
34. Yang, H., Zhu, H., Wang, Y., Huang, M., Shen, Q., Yang, R., Cao, X.: Facescape: A large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (June 2020)
35. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv:2007.08501 (2020)
36. Fan, L.: Image pixelization with differential privacy. In: IFIP Annual Conference on Data and Applications Security and Privacy, Springer (2018) 148–162
37. Ishibuchi, H., Imada, R., Setoguchi, Y., Nojima, Y.: How to specify a reference point in hypervolume calculation for fair performance comparison. Evolutionary computation **26**(3) (2018) 411–440
38. Zheng, T., Chen, C., Yuan, J., Li, B., Ren, K.: Pointcloud saliency maps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 1598–1606
39. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. (1998)
40. Wang, T., Zhang, X., Feng, J., Yang, X.: A comprehensive survey on local differential privacy toward data statistics and analysis. Sensors **20**(24) (2020) 7030
41. Xiao, T., Tsai, Y.H., Sohn, K., Chandraker, M., Yang, M.H.: Adversarial learning of privacy-preserving and task-oriented representations. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 34. (2020) 12434–12441
42. Liu, Z., Wu, Z., Gan, C., Zhu, L., Han, S.: Datamix: Efficient privacy-preserving edge-cloud inference. In: European Conference on Computer Vision, Springer (2020) 578–595
43. McMahan, H.B., Moore, E., Ramage, D., y Arcas, B.A.: Federated learning of deep networks using model averaging. arXiv preprint arXiv:1602.05629 (2016)
44. Gupta, O., Raskar, R.: Distributed learning of deep neural network over multiple agents. Journal of Network and Computer Applications **116** (2018) 1–8
45. Du, J., Li, S., Feng, M., Chen, S.: Dynamic differential-privacy preserving sgd. arXiv preprint arXiv:2111.00173 (2021)
46. Ho, S., Qu, Y., Gu, B., Gao, L., Li, J., Xiang, Y.: Dp-gan: Differentially private consecutive data publishing using generative adversarial nets. Journal of Network and Computer Applications **185** (2021) 103066
47. Jordon, J., Yoon, J., Van Der Schaar, M.: Pate-gan: Generating synthetic data with differential privacy guarantees. In: International conference on learning representations. (2018)
48. Liu, Z., Hu, H., Cao, Y., Zhang, Z., Tong, X.: A closer look at local aggregation operators in point cloud analysis. In: European Conference on Computer Vision, Springer (2020) 326–342
49. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 5589–5598
50. Bytyqi, Q., Wolpert, N., Schömer, E.: Local-area-learning network: Meaningful local areas for efficient point cloud analysis. arXiv preprint arXiv:2006.07226 (2020)
51. Xu, Q., Sun, X., Wu, C.Y., Wang, P., Neumann, U.: Grid-gcn for fast and scalable point cloud learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 5661–5670

52. Xiang, T., Zhang, C., Song, Y., Yu, J., Cai, W.: Walk in the cloud: Learning curves for point clouds shape analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 915–924
53. Lin, C., Li, C., Liu, Y., Chen, N., Choi, Y.K., Wang, W.: Point2skeleton: Learning skeletal representations from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 4277–4286
54. Lang, I., Kotlicki, U., Avidan, S.: Geometric adversarial attacks and defenses on 3d point clouds (2021)
55. Yang, J., Zhang, Q., Fang, R., Ni, B., Liu, J., Tian, Q.: Adversarial attack and defense on point sets (2021)
56. Liu, D., Yu, R., Su, H.: Extending adversarial attacks and defenses to deep 3d point cloud classifiers (2019)
57. Lee, K., Chen, Z., Yan, X., Urtasun, R., Yumer, E.: Shapeadv: Generating shape-aware adversarial 3d point clouds. arXiv preprint arXiv:2005.11626 (2020)
58. Dovrat, O., Lang, I., Avidan, S.: Learning to sample. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2755–2764
59. Gupta, A., Ligett, K., McSherry, F., Roth, A., Talwar, K.: Differentially private combinatorial optimization. In: Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms, SIAM (2010) 1106–1125
60. Feldman, D., Fiat, A., Kaplan, H., Nissim, K.: Private coresets. In: Proceedings of the forty-first annual ACM symposium on Theory of computing. (2009) 361–370