UpCycling: Semi-supervised 3D Object Detection without Sharing Raw-level Unlabeled Scenes

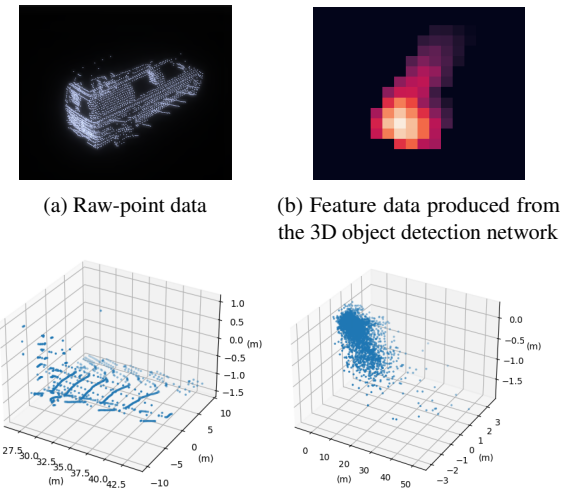
Sunwook Hwang† Youngseok Kim† Seongwon Kim§ Saewoong Bahk† ∗ Hyung-Sin Kim‡ ∗
†Department of Electrical and Computer Engineering, Seoul National University, ∗Corresponding author
§SK Telecom, Seoul, Korea, ‡Graduate School of Data Science, Seoul National University
{swhwang, yskim}@netlab.snu.ac.kr, s1kim@sk.com, {sbahk, hyungkim}@snu.ac.kr

Abstract

Semi-supervised Learning (SSL) has received increasing attention in autonomous driving to reduce the enormous burden of 3D annotation. In this paper, we propose UpCycling*, a novel SSL framework for 3D object detection with zero additional raw-level point cloud: learning from unlabeled de-identified intermediate features (*i.e.*, "smashed" data) to preserve privacy. Since these intermediate features are naturally produced by the inference pipeline, no additional computation is required on autonomous vehicles. However, generating effective consistency loss for unlabeled feature-level scene turns out to be a critical challenge. The latest SSL frameworks for 3D object detection that enforce consistency regularization between different augmentations of an unlabeled raw-point scene become detrimental when applied to intermediate features. To solve the problem, we introduce a novel combination of hybrid pseudo labels and feature-level Ground Truth sampling (F-GT), which safely augments unlabeled multi-type 3D scene features and provides high-quality supervision. We implement* UpCycling *on two representative 3D object detection models: SECOND-IoU and PV-RCNN. Experiments on widely-used datasets (Waymo, KITTI, and Lyft) verify that* UpCycling *outperforms other augmentation methods applied at the feature level. In addition, while preserving privacy,* UpCycling *performs better or comparably to the state-of-the-art methods that utilize raw-level unlabeled data in both domain adaptation and partial-label scenarios.*

1. Introduction

Although the concept of Autonomous Vehicles (AVs) has been around for years, ensuring the safety of users driving AVs on real roads via 3D object detection models is still challenging. To this end, there have been continuous efforts to collect large datasets of 3D road scenes and annotate them carefully [11, 14, 38]. While rapid advances in sensor technology facilitate the collection of 3D scenes at scale, the severe *annotation burden* remains as a main challenge. To alleviate the problem, a couple of semi-supervised learning (SSL) methods for 3D object detection have been proposed recently, such as a combination of perturbation and consistency loss [51] and confidence-based filtering using IoU prediction results [45].



(a) Raw-point data (b) Feature data produced from the 3D object detection network

(c) Original point cloud scene (d) Restored point cloud scene using the inversion attack

Figure 1: Visualization of point cloud scenes. UpCycling improves level of privacy protection since an original point cloud scene cannot be restored from its intermediate feature.

However, these methods learn from unlabeled raw 3D scenes. Collecting a vast amount of raw-level road scenes from AVs can potentially cause disclosure of sensitive private information on the roads [10, 25, 48]. Moreover, the demand for privacy-preserving domains is rapidly accelerating. The EU's General Data Protection Regulation requires firms to implement data protection measures, safeguarding consumers' privacy. This applies even to companies collecting autonomous driving data [27]. In addition, as 2D images can be restored from limited 3D data [28], it's critical to fundamentally secure raw 3D point data.

Given that the problem of potential *privacy leakage* from

raw data collection exists in various applications, a number of studies have tried to not deal with raw data directly. Going beyond encrypting raw data [48], federated learning [12, 16] makes each edge node consume its data locally to train the model and share the model weights (or gradients) instead of raw data. Split learning [12, 36, 42] designs edge nodes to not share raw data but its intermediate feature (*i.e.*, smashed data) that comes from passing through early-stage layers of the model. However, these approaches require local training [32, 35], which makes resource-constrained AVs suffer more *computation overhead*. Given that AVs use significant computing resources to process inference pipelines for 3D detection during driving, such additional computation hinders continuous model updates in natural driving conditions.

In this paper, we aim to address all the three issues: labeling cost, privacy, and AV-side computation overhead. To ensure this end, we propose UpCycling, a novel SSL framework that does not utilize unlabeled raw 3D scenes (Figure 1(a)) but *de-identified, unlabeled* intermediate features (Figure 1(b)) to advance 3D object detection models. Since an unlabeled intermediate feature is naturally produced during a regular detection pipeline with the 3D scene, UpCycling requires neither additional AV-side computation (*e.g.*, local training) nor server-side annotation burden. Further, sharing features instead of raw 3D scenes improves the level of privacy protection as the detection pipeline includes nonlinear layers and compression [5, 19, 33, 49, 53]. Because the process in the nonlinear layers [43] is irreversible, the original scene cannot be completely restored from its intermediate feature. As depicted in Figures 1(c) and (d), the inversion attack [8] attempted on the server side to restore the raw-point data does not result in a successful restoration.[1]

To realize the advantages, UpCycling should provide an effective feature-based SSL method for 3D object detection, which involves two challenges: (1) augmenting unlabeled intermediate features reliably to increase data diversity [15, 18] and (2) providing high-quality pseudo labels to supervise these augmented features. The state-of-the-art (SOTA) semi-supervised 3D object detection frameworks [45, 51] generate consistency loss between weak and strong augmentations of a 3D point scene. However, the augmentation methods targeting raw-level point clouds become detrimental when applied at a feature level. This is because an intermediate feature is a smashed form of its original 3D scene and has multiple types depending on the 3D object detection models, such as grid- and set-types. Therefore, naïve application of the point augmentation methods at a feature level damages the important information in the 3D scene, which causes the pseudo labels to suffer from significant noise.

To address the challenges, we propose high-quality *hybrid pseudo labels* and feature-level ground-truth sampling (*F-GT*). Combining these methods not only achieves significant data diversity but also improves quality of pseudo labels by adding zero-noise labels. We implement UpCycling on two representative 3D detection models, PV-RCNN [33] and SECOND-IoU [40],[2] and perform various experiments on three major datasets for AV applications, KITTI [11], Lyft [14], and Waymo [38]. The results demonstrate the effectiveness of UpCycling in both partial-label and domain adaptation scenarios.

The contributions of this work are summarized as follows:

- UpCycling is the first framework that tackles labeling cost, privacy leakage, and AV-side computation cost altogether to train a 3D object detection model, which deeply investigates how to learn from unlabeled intermediate features.
- UpCycling provides a fresh eye on GT sampling in the context of SSL since it safely improves data diversity of unlabeled feature-level 3D scenes and significantly improves pseudo-label quality by providing zero-noise labels.
- UpCycling not only protects privacy but also performs better or comparably to the SOTA methods in both domain adaptation and partial-label scenarios, on representative models and datasets for 3D object detection.

2. Related Work

Semi-supervised learning. SSL has been actively studied in the context of image classification [18, 26, 37, 39]. Most of the recent SSL methods [15, 18, 26, 39] leverage consistency regularization which trains the model to obtain consistent prediction results across label-preserving data augmentation. In the SSL frameworks, proper data augmentation is essential, which should significantly increase diversity effect without losing consistency with the original data [4, 6]. Accurate pseudo-labeling is another crucial element for SSL to provide high-quality supervision for unlabeled data [20, 37]. While there have been only a couple of studies on SSL for 3D object detection [45, 51], data augmentation and pseudo-labeling are still important. SESS [51] targets indoor 3D object detection, leveraging a teacher-student architecture that takes differently augmented 3D scenes as inputs and utilizes three kinds of consistency losses between outputs. 3DIoUMatch [45] improves quality of pseudo labels with confidence-based filtering in the IoU-guided NMS stage. However, the SSL methods require direct access to a vast amount of raw data, which causes potential privacy leakage.

Feature-level data augmentation. Data diversity can be limited when augmenting only raw data. To further increase diversity, feature-level data augmentation has been investigated [2, 3, 21, 22, 44]. In image classification tasks, adding Gaussian noise to feature-level data gains more data diversity for training and domain generalization [21]. The work

[1] For further details, please refer to Section 5.3 and Supplementary material where more comprehensive information is provided.

[2] SECOND-IoU adds an IoU module to the original SECOND model [49].

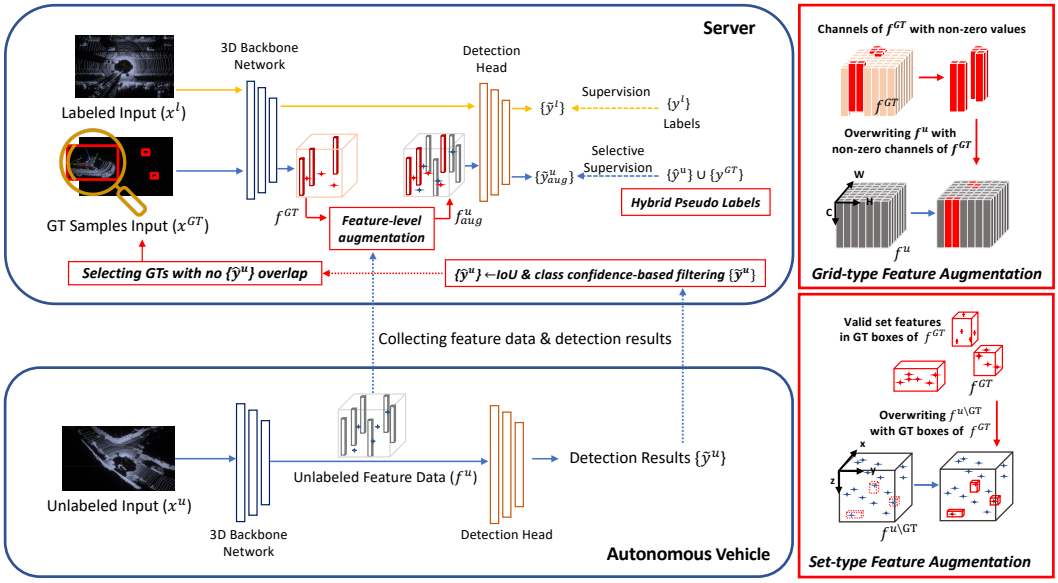


Figure 2: Overview of the UpCycling framework. $\mathbf{f}^u$ and $\{\tilde{\mathbf{y}}^u\}$ refer to unlabeled feature data and detection results from AVs, respectively. IoU and class confidence-based threshold filters detection results to obtain $\{\hat{\mathbf{y}}^u\}$. GTs that do not overlap with $\{\hat{\mathbf{y}}^u\}$ are sampled to form high-quality hybrid pseudo labels. To obtain data diversity, UpCycling augments the collected unlabeled feature-level data $\mathbf{f}^u$ with GT sampling (*F-GT*). The resulting augmented feature, $\mathbf{f}^u_{aug}$, is supervised by the high-quality hybrid pseudo labels.

in [2, 3, 22] resolves lack of data for specific classes by using feature augmentation. Feature augmentation is also applied to few-shot learning in NLP tasks [17]. To our knowledge, however, feature-level augmentation has not been studied in the context of semi-supervised 3D object detection.

Private representation learning. Private representation learning [12, 16] aims to learn from various clients without sharing their raw data, which heavily relies on local training at resource-constrained clients. Federated learning designs clients to not share any data but model weights or gradients with the server. Due to the local computation burden for training the whole model, federated learning methods [24, 31, 46] face significant hurdles in training large neural nets. Split learning [12, 36, 42] is more similar to UpCycling in that clients share intermediate features of local data with the server. However, it still requires local training of early layers of the model. Continuous communication burden during training is another problem of these approaches.

3D object detection models. Main challenges in 3D object detection come from the irregular and sparse positions of 3D point clouds. To address the issues, some researches [29, 34] opt for point-based methods that extract set-type features by processing raw point clouds directly [30]. Other approaches [19, 33, 49, 53] suggest voxel-based methods, which first voxelize a point cloud and extract grid-type features with 3D convolution networks. Therefore, UpCycling should be able to handle both grid- and set-type unlabeled features. Specifically, we adopt two representative 3D object detectors: voxel-based SECOND-IoU [40, 49] and PV-RCNN [33] that mixes point- and voxel-based methods.

3. Method

3.1. Problem Definition

Given a 3D point cloud scene $\mathbf{x}$, we aim to detect a set of 3D bounding boxes and class labels for all objects in $\mathbf{x}$, denoted as $\{\mathbf{y}\}$. We perform this task under a new challenging SSL scenario with unlabeled de-identified data: in contrast to the regular SSL setting, unlabeled raw-level point clouds are not available. Specifically, we have access to N training samples, including N^l labeled point clouds $\{\mathbf{x}^l_i, \{\mathbf{y}^l_i\}\}_{i=1}^{N^l}$ and N^u unlabeled scenes in the form of *intermediate feature* $\{\mathbf{f}^u_i\}_{i=1}^{N^u}$. Here $\mathbf{f}^u$ is the output of the backbone network for an unlabeled point cloud $\mathbf{x}^u$.

3.2. UpCycling Framework

Figure 2 depicts the overall UpCycling framework incorporating server- and AV-side operations. For initialization, the server trains a 3D object detection model on its labeled data $\{\mathbf{x}^l_i, \{\mathbf{y}^l_i\}\}_{i=1}^{N^l}$ and shares the pre-trained model with AVs. UpCycling targets the latest 3D detection models with an IoU module that returns *confidence scores* for bounding box localization. In this paper, we apply UpCycling in PV-RCNN [33] and SECOND-IoU [40]. PV-RCNN is the representative IoU-aware model for 3D object detection and

SECOND-IoU is a modified version of SECOND [49] with addition of IoU module.

For autonomous driving, AVs continuously perform the model's detection pipeline for newly observed 3D scenes. At the same time, to further update the model with more 3D scenes in diverse environments, each AV sends a new 3D scene $\mathbf{x}^u$'s intermediate feature $\mathbf{f}^u$ to the server, which serves as *de-identified unlabeled training data*. It is noteworthy that *zero additional computation* is needed for the de-identification since the feature naturally comes from processing the 3D backbone network in the detection pipeline. Each AV also sends the detection results $\{\tilde{\mathbf{y}}^u\}$ to the server.

With the received features and detection results $\{\mathbf{f}_i^u, \{\tilde{\mathbf{y}}_i^u\}\}_{i=1}^{N^u}$, the server generates consistency loss in a different way of the SOTA SSL methods on 3D object detection that utilize unlabeled raw-point scenes $\{\mathbf{x}_i^u\}_{i=1}^{N^u}$ [45,51]. Specifically, given that supervising $\mathbf{f}^u$ by using its detection result $\{\tilde{\mathbf{y}}^u\}$ again is meaningless, (1) proper augmentation of $\mathbf{f}^u$ and (2) high-quality pseudo labels are essential.

The SOTA methods on semi-supervised 3D object detection [45,51] take a teacher-student architecture [39] by using random sampling (RS) for weak augmentation and both RS and Flip for strong augmentation of a point cloud. However, in our scenario where an input is an intermediate feature, the augmentation methods significantly damage the original scene. Instead, we propose feature-level ground-truth sampling (*F-GT*) for feature augmentation, as illustrated in Figure 2. Although ground-truth (GT) sampling has been used as a point cloud augmentation method for supervised 3D object detection [5, 19, 33, 49, 53] and is known to provide at most fair performance improvement [13], we claim that its impact can be more significant when it comes to *feature-level* augmentation of an *unlabeled* 3D scene. This is because *F-GT* tackles one of the most crucial issues for successful SSL: improving the quality of pseudo labels for unlabeled features by generating *hybrid pseudo labels*.

3.3. Hybrid Pseudo Labels

For effective SSL, we adopt *F-GT* to augment an unlabeled scene feature $\mathbf{f}^u$ and include the sampled GT labels (zero-noise labels) in the pseudo-label set for the unlabeled feature. By doing so, UpCycling constructs high quality *hybrid pseudo labels*.

Confidence-based pseudo-label filtering. First, inspired by 3DIoUMatch [45], UpCycling screens the received detection results $\{\tilde{\mathbf{y}}^u\}$ by using each $\tilde{\mathbf{y}}^u$'s confidence scores for both object classification and bounding box localization. Assume that τ_{IoU} and τ_{cls} are thresholds for box localization and object classification, respectively. UpCycling filters out a detection result if its class confidence or localization confidence is lower than the given threshold, leaving a set of high-quality pseudo labels, denoted as $\{\hat{\mathbf{y}}^u\}$. The confidence-based pseudo-label filtering is applied for more accurate supervision.

Pseudo-label-aware GT sampling. When GT sampling is applied for supervised learning, it first constructs a GT database that consists of labeled 3D bounding boxes and point clouds in the boxes, collected from the entire labeled training set $\{\mathbf{x}_i^l, \{\mathbf{y}_i^l\}\}_{i=1}^{N^l}$. To augment a labeled 3D scene $\mathbf{x}^l$, GTs are sampled from the database and randomly placed in the 3D scene. To avoid tampering with GT information, a GT sample that overlaps with a ground-truth bounding box in the original labeled scene is removed.

In contrast, our *F-GT* aims to augment an *unlabeled* 3D scene feature $\mathbf{f}^u$ without accurate box labels. Instead, given that a set of high-quality pseudo labels $\{\hat{\mathbf{y}}^u\}$ is provided, *F-GT* samples GTs that do not overlap with the *pseudo labels*. Importantly, although the pseudo labels are filtered with the two thresholds τ_{IoU} and τ_{cls}, these thresholds are set moderately [45], enabling the pseudo labels to cover most objects in the original scene $\mathbf{x}^u$; GT samples are likely to be placed on the background of $\mathbf{x}^u$.

Hybrid pseudo-labels. To generate pseudo labels that supervise an augmented unlabeled feature $\mathbf{f}_{aug}^u$, UpCycling merges the high-quality pseudo-label set for the original feature $\mathbf{f}^u$, $\{\hat{\mathbf{y}}^u\}$, with the label set for the GT samples, $\{\mathbf{y}^{GT}\}$, resulting in a set of *hybrid pseudo labels* $\{\hat{\mathbf{y}}^u\} \cup \{\mathbf{y}^{GT}\}$. Given that $\{\mathbf{y}^{GT}\}$ are literally ground-truth labels with *zero noise*, adding these labels to the pseudo labels enables powerful supervision. Furthermore, generating the hybrid pseudo labels does not need to execute the inference pipeline at the server, since all GT labels are already given.

3.4. Feature-level 3D Scene Augmentation

Regarding *F-GT*, since the server does not have an original unlabeled scene $\mathbf{x}^u$ but only its intermediate feature $\mathbf{f}^u$, it is impossible to directly place GT samples on the point cloud scene. Instead, *F-GT* generates a separate point cloud input that comprises only GT samples. The GT-only point cloud passes through the model's 3D backbone network, resulting in a GT-only feature $\mathbf{f}^{GT}$. Note that while the 3D backbone of SECOND-IoU generates only grid-type features, that of PV-RCNN [33] generates both grid- and set-type features. To this end, *F-GT* augments $\mathbf{f}^u$, grid- or set-type feature, as follow:

Grid-type feature augmentation. As shown in Figure 2, when $\mathbf{f}^u$ and $\mathbf{f}^{GT}$ are grid-type features, *F-GT* generates an augmented feature by overwriting $\mathbf{f}^u$ with $\mathbf{f}^{GT}$; if a channel on $\mathbf{f}^{GT}$ has non-zero values, the $\mathbf{f}^{GT}$ channel replaces that in $\mathbf{f}^u$. Giving higher priority for $\mathbf{f}^{GT}$ removes some information included in $\mathbf{f}^u$. However, given that the GT samples take up a tiny portion of an entire scene (*i.e.*, most values in $\mathbf{f}^{GT}$ are zero), only a small number of values in $\mathbf{f}^u$ are modified. In addition, the removed information in $\mathbf{f}^u$ is related to the background since the sampled GTs are not overlapped with pseudo labels, which does not harm model training.

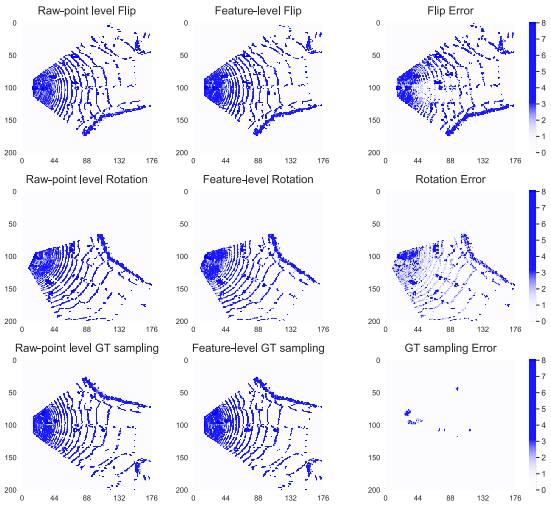


Figure 3: Feature-level scenes for three data augmentation methods: Flip (1st row), Rotation (2nd row), and GT sampling (3rd row). Feature-level scenes of raw-point level augmentation are on the left. Feature-level scenes of feature-level augmentation are in the middle. Heatmaps of RMSE based on comparison between raw-level and feature-level augmentation scenes are on the right.

Set-type feature augmentation. When an unlabeled feature $\mathbf{f}^u$ and a GT sample feature $\mathbf{f}^{GT}$ are set types, each of them consists of n represented points, denoted as $\mathbf{f}^u = \{f_i^u\}_{i=1}^n$ and $\mathbf{f}^{GT} = \{f_i^{GT}\}_{i=1}^n$, respectively. In this case, as illustrated in Figure 2, *F-GT* generates an augmented feature as a point set, denoted as $\mathbf{f}_{aug}^u = \{f_{aug,i}^u\}_{i=1}^n$. To this end, we first exclude the scene feature points f_i^u that are in the GT boxes, generating $\mathbf{f}^{u\backslash GT}$. Then each feature point $f_{aug,i}^u$ is randomly sampled from either $\mathbf{f}^{u\backslash GT}$ or $\mathbf{f}^{GT}$.

In doing so, it is important that the scene feature contains much more information than the GT feature; for reasonable augmentation, $\mathbf{f}_{aug}^u$ should include scene feature points more than GT feature points. To determine proper sampling frequency, we utilize the information in the grid-type feature that is generated simultaneously with the set-type feature by the 3D backbone network: how many values in the grid-type feature for the scene and GT samples are non-zero. For example, if the number of grid with non-zero values in the scene and GT features (grid types) is 2000 and 50, respectively, points in the augmented feature set $\mathbf{f}_{aug}^u$ is sampled from $\mathbf{f}^{u\backslash GT}$ 400 times more than $\mathbf{f}^{GT}$.

3.5. Loss

The model's detection head is trained to predict the hybrid pseudo labels for the augmented feature $\mathbf{f}_{aug}^u$. Given that our target models have an IoU module as well as a Region Proposal Network (RPN), the unlabeled loss $\mathcal{L}(\mathbf{f}_{aug}^u)$

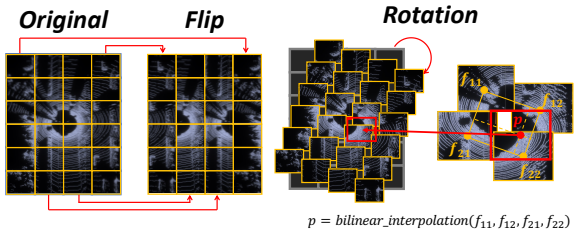


Figure 4: Conceptual images of feature-level augmentation with Flip and Rotation.

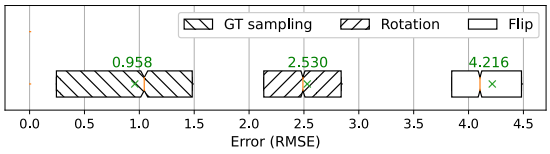


Figure 5: RMSE between raw- and feature-level augmentations of the entire KITTI training dataset. Box range covers the first quartile to the third quartile and the mark '×' indicates the mean value.

includes loss of each of the two modules as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{f}_{aug}^u) =& \mathcal{L}_{loc}^{RPN}(\{\hat{\mathbf{y}}^u\} \cup \{\mathbf{y}^{GT}\}) + \mathcal{L}_{loc}^{IoU}(\{\hat{\mathbf{y}}^u\} \cup \{\mathbf{y}^{GT}\}) \\ &+ \mathcal{L}_{cls}^{RPN}(\{\hat{\mathbf{y}}^u\} \cup \{\mathbf{y}^{GT}\}).\end{aligned} \tag{1}$$

The exact calculation of the three terms depends on the model architecture, following the calculation of supervised loss. Assuming that a training batch consists of a set of labeled scenes $\{\mathbf{x}^l\}$ and a set of augmented features for unlabeled scenes $\{\mathbf{f}_{aug}^u\}$, the total loss for the batch is calculated as below, where w is the unsupervised loss weight:

$$\mathcal{L}_{total} = \mathcal{L}(\{\mathbf{x}^l\}) + w\mathcal{L}(\{\mathbf{f}_{aug}^u\}). \tag{2}$$

4. Analysis on 3D Scene Feature Augmentation

In this section, we take a deeper look into subtle feature-level 3D scene augmentation. Specifically, we focus on why widely-used point cloud augmentation methods damage important information when applied at a feature level.

To this end, Figure 3 depicts activation heat maps of the Bird-eye View (BEV) compression module in SECOND-IoU when Flip, Rotation, and GT sampling are applied to an example 3D scene covering x, y, z axis range 70.4, 80, 4 meters. The figure shows that in the cases of Flip and Rotation, raw-level augmentation (*i.e.*, flipping/rotating the whole point cloud) and feature-level augmentation (*i.e.*, flipping/rotating the feature vector) result in significantly different activations. In both cases, although the two activation heat maps look similar at a glance, taking the difference between the two causes errors that are widely spread over the entire feature map. In contrast, when using GT sampling,

raw- and feature-level augmentations provide similar activation heat maps. Although some errors exist, they are placed in restricted areas where GT samples are inserted.

Figure 4 provides a visual illustration of Flip and Rotation for feature augmentation. If a point cloud is voxelized with each voxel producing its feature value, flipping/rotating the feature vector is similar to flipping/rotating voxels. This means that point locations are shifted not individually but in groups, and the geometric relationship between intra-voxel points is maintained; they are neither flipped nor rotated. In the worst case, the group (voxel)-wise flipping causes a valid car object to break apart, making its label detrimental to training. Breaking the geometric relationship between points on the background can also cause severe misinterpretation. Similarly, the group-wise rotation breaks the geometric relationship mildly and its bilinear interpolation creates the errors, which is not proper for augmentation.

Figure 5 confirms our description by showing the average of root mean square error (RMSE) between raw- and feature-level augmentations in the KITTI dataset. This plot illustrates that feature-level Flip and Rotation severely damage the original scene, in contrast to GT sampling, which only produces minor errors.

5. Experiments

5.1. Experimental Setup

Scenarios. To demonstrate the effectiveness of UpCycling in various practical situations, we conduct experiments in both domain adaptation and partial-label scenarios. The domain adaptation task is to adapt the model, which is trained on abundant labeled data in the source domain, to an unseen target domain that provides only unlabeled data. In the partial-label scenario, the model is trained and tested in the same domain but most of the training data is unlabeled.

Datasets. We choose three datasets widely used for detection applications of AVs: Waymo [38], Lyft [14], and KITTI [11]. Among the three, the Waymo dataset is the most diverse and the largest in volume. The 3D scenes in the Waymo dataset are captured in Phoenix, Mountain View, and San Francisco, the US, under multiple weather and time settings. The Lyft dataset is collected around Palo Alto, the US, in clear weather in the daytime. The KITTI dataset is collected in Karlsruhe, Germany, in clear weather during the daytime. Due to regional characteristics, car sizes in KITTI are different from those in Waymo and Lyft [47]. We focus on car objects in this section and more details are in the supplementary material.

Implementation details. When training a model with UpCycling, we set the two filtering thresholds τ_{IoU} and τ_{cls} to 0.5 and 0.4, respectively, and the weight for the loss $\mathcal{L}(\{f^u_{aug}\})$ is set as $w = 1$. We set the ratio of labeled data to unlabeled data in a mini-batch to 1:2 and 1:1 for domain adaptation and partial-label experiments, respectively. Importantly, *F-GT* samples GT boxes only from the labeled dataset: the source domain data in the domain adaptation scenario and a small portion of labeled data in the partial-label scenario. Lastly, UpCycling freezes the 3D backbone network after training it on the labeled data to prevent the divergence between an intermediate feature from the server's 3D backbone network and that collected from AVs. Therefore, UpCycling updates only the detection head using unlabeled feature-level data. More details are in the supplementary material.

Table 1: Effects of feature augmentation methods in a partial-label scenario where the 3D object detection model is SECOND-IoU and 10% training data is labeled in KITTI.

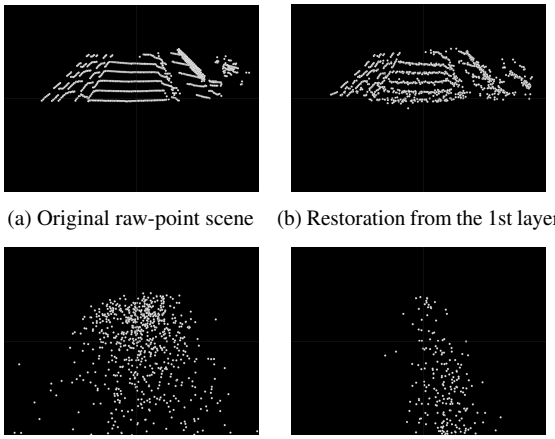
Policy #	*Flip*	*Noise*	*RS*	*Scale*	*Rot.*	*F-GT*	AP_{3D} Easy	Mod	Hard
Baseline							70.58	56.00	47.94
1	✓						−16.31	−20.09	−19.79
2		✓					+0.03	+0.13	−1.23
3			✓				+2.47	−0.96	+0.63
4∗	✓		✓				−11.69	−13.75	−13.32
5				✓			−15.16	−13.94	−11.50
6					✓		+4.80	+5.42	+7.96
UpCycling						✓	**+7.81**	**+7.87**	**+8.14**

5.2. Effect of Feature Augmentation Schemes

First, we investigate feature augmentation deeply by evaluating the superiority of *F-GT*, which is utilized for UpCycling, to other augmentation schemes in a partial-label scenario. To this end, we train SECOND-IoU on the KITTI dataset when only 10% of its training data is labeled. Importantly, given that the KITTI dataset is originally shuffled regardless of place and time sequence, we rearrange it in chronological order for each place to prevent the data leakage between the labeled and unlabeled sets [1].

Comparison schemes. In this scenario, **Baseline** trains the model using only the limited amount of labeled data. **Flip** and **RS** are used in the SOTA SSL methods on 3D object detection to augment raw-level 3D scenes [45, 51]. For feature-level Flip, we place feature information to its symmetric position on the feature map. For feature-level RS, we nullify randomly selected 5% of feature data. Combination of feature-level Flip and RS is actually a feature-level variant of the SOTA 3DIoUMatch [45], named **F-3DIoUMatch**.[3] **Noise** is an existing feature augmentation method that adds Gaussian noise, which is used for domain generalization of image classification [21]. **Scale** refers to the random adjustment of the overall size of a feature scene. When we scale the feature scene, the scaling factor is randomly selected from the range [0.95, 1.05]. Lastly, **Rotation** rotates the feature with a degree randomly selected from [-45°, 45°] and performs bilinear interpolation.

[3] Policy 4∗ indicates F-3DIoUMatch.



(a) Original raw-point scene (b) Restoration from the 1st layer

(c) Restoration from a middle (3rd) layer (d) Restoration from the last (5th) layer, same as UpCycling

Figure 6: Results of inversion attack for the 3D backbone model (5 convolutional layers) of SECOND-IoU and PV-RCNN. The example 3D point cloud scene is in KITTI.

We conducted **additional experiments** on feature-level Random Scaling, which reveals a **considerable decrease** of 13.9 in AP_{3D}@Mod compared to the baseline model. We will mention Random Scaling with the experimental results.

Result analysis. Table 1 shows each augmentation scheme's performance margin compared to Baseline in the partial-label scenario. Flip and Scale significantly underperform Baseline despite the use of much more (unlabeled) training data, verifying that these feature-level augmentations damage important information in 3D scenes. Both Noise and RS have marginal impact on performance, showing that these perturbation strategies do not result in meaningful data diversity. Combining Flip and RS (*i.e.*, F-3DIoUMatch) still performs worse than Baseline due to the negative effect of Flip, which confirms that naïve application of SOTA SSL methods at a feature level does not work. Although Rotation improves performance, our *F-GT* provides the lowest augmentation errors (Figure 5) and thus *the best performance* in all cases.

5.3. Privacy Protection of Feature Sharing

As neural network activations could be inverted to reconstruct input data [9, 23, 41], there could be concerns on potential privacy leaks when sharing features. We investigate whether an inversion attack can recover the grid-type feature data generated from both the SECOND-IoU and PV-RCNN backbone networks to the original point cloud. To this end, we implement the inversion attack model using the decoder method [8] that is widely used to evaluate whether a model consisting of convolutional layers can be inverted [7, 52].[4]

[4]To the best of our knowledge, there has been no research that particularly focuses on inversion attacks for 3D point clouds.

Table 2: Domain adaptation results with two target datasets: KITTI and Lyft. Difficulty of the KITTI test dataset is set as Moderate. Baseline is a pre-trained model with Waymo whereas Oracle is trained with fully labeled target dataset.

Dataset	Method	SECOND-IoU	PV-RCNN
		AP_{BEV} / AP_{3D}	AP_{BEV} / AP_{3D}
Lyft	Baseline	30.20 / 21.32	33.00 / 24.49
	SN	28.38 / 19.25	33.44 / 25.64
	ST3D	60.53 / 29.90	62.28 / 42.63
	UpCycling	**68.83 / 45.66**	**63.38 / 46.83**
	ST3D (w/ SN)	52.86 / 21.25	60.15 / 44.02
	UpCycling (w/ SN)	**65.10 / 49.24**	**63.58 / 49.35**
	Oracle	76.70 / 61.70	78.68 / 64.54
KITTI	Baseline	54.14 / 10.16	62.24 / 9.24
	SN	60.80 / 37.30	60.08 / **38.86**
	ST3D	**70.90 / 40.16**	**66.19** / 23.26
	UpCycling	58.26 / 11.71	62.09 / 11.35
	ST3D (w/ SN)	80.97 / 57.68	54.30 / 48.79
	UpCycling (w/ SN)	**84.12 / 67.65**	**85.90 / 61.12**
	Oracle	90.36 / 82.02	90.84 / 84.56

More details are in the supplementary material.

Result analysis. We conduct an inversion attack on the 3D backbone network in SECOND-IoU and PV-RCNN.[5] Figures 6(b)-(d) present the restoration results for intermediate features at three different convolutional layers of the backbone network: 1st, 3rd, and 5th (last) layers, respectively. While the restored point cloud from the first layer is relatively similar to the original scene (Figure 6(b)), it becomes significantly different when applied to deeper layers' features (Figures 6(c) and (d)). As the number of nonlinear layers increases, it becomes more difficult to accurately restore the original data. Furthermore, restoring a point cloud from its intermediate feature is particularly challenging since each raw point needs to be positioned precisely in voxelized spaces. UpCycling utilizes unlabeled features at the last (deepest) layer, making it impossible to accurately recover the original scene from an intermediate feature. Supplementary material contains more inversion examples.

5.4. Domain Adaptation Experiments

Although UpCycling offers privacy protection by using only intermediate features, it is crucial to evaluate whether it provides competitive detection accuracy compared to the SOTA methods that use raw-level point clouds (Sections 5.4 and 5.5). In domain adaptation experiments, we use the Waymo dataset as the source domain and the Lyft and KITTI datasets as the target domains. The model is first pre-trained on the source domain's labeled data (called the baseline model), adapted using unlabeled training data in a target domain, and then tested on the target domain's test data.

Comparison schemes. We compare UpCycling with various

[5]The 3D point cloud scene in Figure 6(a) is from KITTI dataset, and the point cloud range covers the x, y, and z-axis ranges 17.6, 20, and 4 meters.

Table 3: Partial-label scenario results with three portions of labeled data in the KITTI dataset: 2%, 10%, 25%.

AP_{3D}		2%			10%			25%		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
SECOND-IoU	Baseline	56.69	44.11	37.19	70.58	56.00	47.94	84.47	71.06	62.87
	3DIoUMatch	63.57	49.58	43.00	71.76	57.01	50.08	81.71	68.51	60.92
	improved (%)	12.13	12.39	15.62	1.67	1.80	4.47	-3.26	-3.59	-3.11
	UpCycling	70.19	59.97	44.83	76.09	60.41	51.84	85.22	72.87	63.93
	improved (%)	**23.81**	**35.96**	**20.54**	**7.81**	**7.87**	**8.14**	**0.89**	**2.55**	**1.69**
PV-RCNN	Baseline	68.10	53.27	46.20	81.23	68.67	60.32	87.63	76.03	68.62
	3DIoUMatch	81.04	65.77	58.83	85.26	70.64	63.32	85.08	72.37	65.02
	improved (%)	**19.00**	**23.47**	**27.34**	**4.97**	**2.87**	4.98	-2.91	-4.81	-5.25
	UpCycling	76.46	61.44	52.94	83.64	69.60	63.53	88.05	76.61	70.80
	improved (%)	12.28	15.34	14.59	2.97	1.35	**5.32**	**0.48**	**0.76**	**3.18**

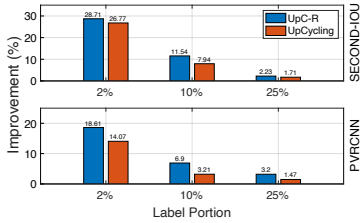


Figure 7: UpC-R vs. UpCycling: Partial-label results in the KITTI dataset. The average performance improvement in all KITTI test cases (easy, moderate, and hard).

methods. **Baseline** evaluates the baseline model directly and **Oracle** adapts the model with fully supervised learning in the target domain, which provide the lower- and upper-bound performance, respectively. **ST3D** [50] and **SN** (Statistical Normalization) [47] are the SOTA domain adaptation methods on 3D object detection that utilize unlabeled raw 3D scenes. ST3D generates pseudo labels from unlabeled data in the target domain to adapt the baseline model. SN assumes that statistical object sizes in the target domain are given and trains the baseline model in the source domain using the target domain object size information. We also evaluate variants of ST3D and our UpCycling by combining SN together, denoted as **(w/ SN)**.

Result analysis. Table 2 shows the results of UpCycling and the various comparison methods on SECOND-IoU and PV-RCNN. Surprisingly, the results show that although UpCycling (or w/ SN) does not utilize raw-point scenes for privacy protection, it *provides the best accuracy* in most cases. Specifically, UpCycling (or w/ SN) significantly outperforms the two SOTA methods (ST3D and SN) in the Lyft case. When compared to the better option between ST3D (or w/ SN) and SN in each case, UpCycling improves accuracy by **1.3**$\sim$**19.71** AP_{BEV} and **5.33**$\sim$**19.34** AP_{3D}. The results demonstrate the effectiveness of hybrid pseudo labels and feature-level augmentation schemes in UpCycling and also suggest the potential of using unlabeled features to advance 3D object detection models.

Taking a deeper look, SN significantly improves UpCycling performance in the KITTI dataset. Since object sizes in KITTI are different from those in Lyft and Waymo, adjusting object sizes with SN for UpCycling is effective.

5.5. Partial-label Experiments

In partial-label experiments, we use the same setting as in Section 5.2 but train both SECOND-IoU and PV-RCNN.

Comparison schemes. In this scenario, **Baseline** trains the model using only the limited amount of labeled data. **3DIoUMatch** [45] is the SOTA SSL method using unlabeled raw-point scenes. For consistency regularization, 3DIoUMatch uses Flip and RS to augment raw data and filters pseudo labels in the IoU-guided NMS.[6]

Result analysis. Table 3 shows that UpCycling outperforms 3DIoUMatch in most cases by effectively utilizing unlabeled feature-level data. In the case of 25%, 3DIoUMatch even underperforms Baseline but UpCycling maintains performance improvement on both SECOND-IoU and PV-RCNN. The results are interesting because the scenario is unfavorable for UpCycling in that (1) UpCycling trains the 3D backbone only using the small portion of labeled data and (2) the effect of *F-GT* could be marginal since the number of GT samples are proportional to that of labeled data. UpCycling successfully overcomes the disadvantages, verifying that it achieves significant performance improvement even when using a relatively immature backbone network and *F-GT* effectively augments a large number of unlabeled data when only a small number of GTs are available.

5.6. Ablation Studies

Since UpCycling freezes the backbone during the SSL process for effective feature sharing, we evaluate the effect of the backbone freezing. To this end, we devise a comparison scheme UpC-R, the application of UpCycling at the raw-level input. UpC-R augments a raw-level 3D scene using GT samples and trains the whole network including the backbone using unlabeled data and hybrid pseudo labels. Note that this approach not only sacrifices privacy but also takes much longer to train compared to UpCycling.

Result analysis. Figure 7 compares UpC-R and UpCycling in the partial-label scenario in Section 5.5. While sacrificing privacy, UpC-R outperforms UpCycling by training the backbone further. Interestingly, UpC-R performs even better than the SOTA 3DIoUMatch (Table 3), demonstrating that GT sampling is more effective augmentation than the combination of Flip and RS *even at the raw-input level*. On the other hand, the performance gap between UpC-R and UpCycling decreases as the number of labeled data increases, meaning that once the backbone is well-trained, the combination of

[6]Since the authors in [45] did not use the rearranged KITTI dataset in their experiments, we measure the performance of 3DIoUMatch again in the rearranged KITTI dataset. In addition, we newly implement 3DIoUMatch on SECOND-IoU for more extensive comparison.

hybrid pseudo-labels and GT-based augmentation can be applied flexibly to any layer without performance degradation. We see this as the unique advantage of GT sampling that other point cloud augmentation methods cannot provide.

6. Conclusion

In this paper, we present that UpCycling, a novel semi-supervised learning method for 3D object detection models, improves model performance by gathering de-identified unlabeled data from AVs. To the best of our knowledge, no study has considered labeling cost, privacy, and edge computing resources in AVs and overall systems altogether. Taking all these factors into account, we apply UpCycling to the representative 3D object detection models, SECOND-IoU and PV-RCNN. Through various experiments using multiple datasets, we verify the superiority of UpCycling in partial-label situations as well as domain adaptation in comparison with other SOTA methods. Furthermore, we also confirm that feature-level GT sampling can improve model performance significantly compared with other augmentation methods applied at a feature level.

Acknowledgement

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center support program (IITP-2023-2021-0-02048) supervised by the Institute for Information & Communications Technology Planning & Evaluation, and the National Research Foundation (NRF) of Korea grant funded by the Korea government (MSIT) (No. 2022R1A5A1027646) and (No. RS-2023-00212780).

References

[1] bostondiditeam. Exploratory findings for the kitti vision benchmark suite. https://github.com/bostondiditeam, 2017. 6

[2] Feng Cen, Xiaoyu Zhao, Wuzhuang Li, and Guanghui Wang. Deep feature augmentation for occluded image classification. *Pattern Recognition*, 111:107737, 2021. 2, 3

[3] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, pages 694–710. Springer, 2020. 2, 3

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 2

[5] Jiajun Deng, Shaoshuai Shi, Peiwei Li, W. Zhou, Yanyong Zhang, and H. Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI*, 2021. 2, 4

[6] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2

[7] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 7

[8] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 7

[9] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837, 2016. 7

[10] David Eckhoff and Christoph Sommer. Driving for big data? privacy concerns in vehicular networking. *Security & Privacy, IEEE*, 12:77–79, 01 2014. 1

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 6

[12] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018. 2, 3

[13] Martin Hahner, Dengxin Dai, Alexander Liniger, and Luc Van Gool. Quantifying data augmentation for lidar based 3d object detection. *arXiv preprint arXiv:2004.01643*, 2020. 4

[14] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. *CoRR*, abs/2006.14480, 2020. 1, 2, 6

[15] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d 'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 10759–10768. Curran Associates, Inc., 2019. 2

[16] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency, 2016. 2, 3

[17] Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. A closer look at feature space data augmentation for few-shot intent classification. *arXiv preprint arXiv:1910.04176*, 2019. 3

[18] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning, 2016. 2

[19] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, June 2019. 2, 3, 4

[20] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013. 2

[21] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for

domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021. 2, 6

[22] Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3

[23] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015. 7

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR. 3

[25] Yang Ming and Xiaopeng Yu. Efficient privacy-preserving data sharing for fog-assisted vehicular sensor networks. *Sensors*, 20(2), 2020. 1

[26] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019. 2

[27] Trix Mulder and Nynke E Vellinga. Exploring data protection challenges of automated driving. *Computer Law & Security Review*, 40:105530, 2021. 1

[28] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 145–154, 2019. 1

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

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[31] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020. 3

[32] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2021–2031. PMLR, 26–28 Aug 2020. 2

[33] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3, 4

[34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[35] Nir Shlezinger, Mingzhe Chen, Yonina C. Eldar, H. Vincent Poor, and Shuguang Cui. Uveqfed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing*, 69:500–514, 2021. 2

[36] Abhishek Singh, Praneeth Vepakomma, Otkrist Gupta, and Ramesh Raskar. Detailed comparison of communication efficiency of split learning and federated learning. *arXiv preprint arXiv:1909.09145*, 2019. 2, 3

[37] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 2

[38] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 6

[39] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, 2017. 2, 4

[40] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 2, 3

[41] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018. 7

[42] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018. 2, 3

[43] Praneeth Vepakomma, Tristan Swedish, Ramesh Raskar, Otkrist Gupta, and Abhimanyu Dubey. No peek: A survey of private distributed deep learning. *arXiv preprint arXiv:1812.03288*, 2018. 2

[44] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR, 09–15 Jun 2019. 2

[45] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14615–14624, 2021. 1, 2, 4, 6, 8

[46] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020. 3

[47] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 6, 8

[48] Jinbo Xiong, Renwan Bi, Mingfeng Zhao, Jingda Guo, and Qing Yang. Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles. *IEEE Wireless Communications*, 27(3):24–30, 2020. 1, 2

[49] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, Oct 2018. 2, 3, 4

[50] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2021. 8

[51] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self-ensembling semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11079–11087, 2020. 1, 2, 4, 6

[52] Xuejun Zhao, Wencan Zhang, Xiaokui Xiao, and Brian Lim. Exploiting explanations for model inversion attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 682–692, October 2021. 7

[53] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2, 3, 4